

Міністерство науки та освіти України
Національний технічний університет України «Київський політехнічний інститут»

Розрахункова робота по курсу „КЛ-2”

Виконав

студент І-го курсу ФІОТ гр. ІО-93
Білогородський Владислав Юрійович
Керівник: Жабін В.І.

Київ 2010

ЗМІСТ

Завдання та обґрунтування варіанту.....	5
1 Операція множення чисел.....	6
1.1 Перший спосіб множення.....	7
1.1.1 Теоретичне обґрунтування способу.....	7
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'ять.....	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.....	11
1.2.4 Таблиця станів регістрів.....	12
1.2.5 Функціональна схема пристрою.....	12
1.2.6 Закодований мікроалгоритм.....	13
1.2.7 Граф управляючого автомата.....	13
1.2.8 Обробка порядків	14
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.....	16
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	20
1.3.9 Форма подання результату в пам'яті.....	20
1.4 Четвертий спосіб множення.....	21
1.4.1 Теоретичне обґрунтування способу.....	21
1.4.2 Операційна схема.....	21
1.4.3 Змістовний мікроалгоритм.....	21
1.4.4 Таблиця станів регістрів.....	22
1.4.5 Функціональна схема пристрою.....	22
1.4.6 Закодований мікроалгоритм.....	23
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.....	25
2.1 Перший спосіб.....	25
2.1.1 Теоретичне обґрунтування способу.....	25
2.1.2 Операційна схема.....	25
2.1.3 Змістовний мікроалгоритм.....	26

2.1.4 Таблиця станів реєстрів.....	26
2.1.5 Функціональна схема пристрою.....	27
2.1.6 Закодований мікроалгоритм.....	28
2.1.7 Граф управляючого автомата.....	28
2.1.8 Обробка порядків	28
2.1.9 Форма подання результату в пам'яті.....	28
2.2 Другий спосіб.....	29
2.2.1 Теоретичне обґрунтування способу.....	29
2.2.2 Операційна схема.....	29
2.2.3 Змістовний мікроалгоритм.....	30
2.2.4 Таблиця станів реєстрів.....	30
2.2.5 Функціональна схема пристрою.....	31
2.2.6 Закодований мікроалгоритм.....	32
2.2.7 Граф управляючого автомата.....	32
2.2.8 Обробка порядків	33
2.2.9 Форма подання результату в пам'яті.....	33
3 Операція додавання чисел	
3.1 Теоретичне обґрунтування способу.....	34
3.2 Операційна схема.....	34
3.3 Змістовний мікроалгоритм.....	35
3.4 Таблиця станів реєстрів.....	35
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.....	37
4 Операція додавання чисел	
4.1 Теоретичне обґрунтування способу.....	38
4.2 Операційна схема.....	38
4.3 Змістовний мікроалгоритм.....	39
4.4 Таблиця станів регістрів.....	39
4.5 Функціональна схема пристрою.....	40
4.6 Закодований мікроалгоритм.....	41
4.7 Граф управляючого автомата.....	41
4.8 Обробка порядків	41
4.9 Форма подання результату в пам'яті.....	41
5 Синтез управляючого автомата для операційного пристрою (дати назву пристрою)	
5.1 Таблиця співвідношення управляючих входів операційного автомата і виходів управляючого автомата	42
5.2 Мікроалгоритм в термінах управляючого автомата.....	42
5.3 Структурна таблиця автомата.....	43
5.4 Синтез функцій виходів і переходів.....	43
5.5 Функціональна схема пристрою (виходи управляючого автомата підключені до входів операційного автомата).....	45

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). (0,5)
- 2) Виконати 8 операцій з числами X_2 і Y_2 з плаваючою комою (чотири способи множення, два способи ділення, додавання та добування кореня з X_2). Номери операцій (для п.3) відповідають порядку переліку (наприклад, 6 – ділення другим способом).

Для обробки мантис кожної операції, подати:

- 2.1) теоретичне обґрунтування способу; (0,2)
 - 2.2) операційну схему; (0,2)
 - 2.3) змістовний мікроалгоритм; (0,2)
 - 2.4) таблицю станів регістрів (лічильника), довжина яких забезпечує одержання 15 основних розрядів мантиси результату; (1,5)
 - 2.5) функціональну схему з відображенням управляючих сигналів; (0,5)
 - 2.6) закодований мікроалгоритм (мікрооперації замінюються управл. сигналами); (0,3)
 - 2.7) граф управляючого автомата Мура з кодами вершин; (0,5)
 - 2.8) обробку порядків (показати у довільній формі); (0,5)
 - 2.9) форму запису нормалізованого результату з плаваючою комою в пам'ять. (0,1)
- Операцію додавання до етапу нормалізації результату можна проілюструвати у довільній формі. Вказані пункти виконати для етапу нормалізації результату з урахуванням можливого нулевого результату.
- 3) Для операції з номером $x_3x_2x_1$ побудувати управляючий автомат Мура на тригерах (тип вибрати самостійно) і елементах булевого базису. (1,5)

Обґрунтування варіанту

9301=10010001010111

$X_2 = -11111011,01111$
 $Y_2 = +11111,0110001111$

Запис чисел X_2 і Y_2

$X_{2\text{ПК}} = 1.11111011,1100100$
 $Y_{2\text{ПК}} = 0.11111,0111100001$

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

ЗН	М														ЗН	Р							
1	1	1	1	1	1	0	1	1	0	0	0	1	1	1	1	0	0	0	0	1	0	0	0

$C_x = 8 - 2^7 = -120 = 10001000$

3H	M														C								
1	1	1	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0

$P_y = 5_{10} = 101_2$

ЗН	М														ЗН	Р							
0	1	1	1	1	1	0	1	1	0	0	0	1	1	1	1	0	0	0	0	0	1	0	1

$C_y = 5 - 2^7 = -123 = 10000101$

ЗН	М														С								
0	1	1	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	0	0	1	0	1

1 Операції множення чисел

1.1 Перший спосіб множення

1.1.1 Вираз $Z = X \odot Y = Yx_12^{-1} + Yx_22^{-2} + \dots + Yx_i2^{-i} + \dots + Yx_n2^{-n}$ можна подати у вигляді:

$Z_i = (Z_{i-1} + Yx_{n-i+1})2^{-1}$ з початковими значеннями $i = 1, Z_0 = 0$, причому $Z_n = Z = Y * X$. У розглянутому способі множення здійснюється з молодших розрядів множника, сума часткових добутоків зсувається вправо, а множене залишається нерухомим.

1.1.2 Операційна схема:

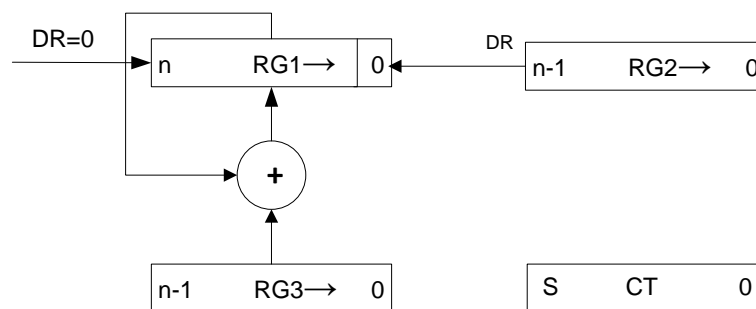


Рис. 1.1 – Операційна схема множення I способом

1.1.3 Змістовний мікроалгоритм:

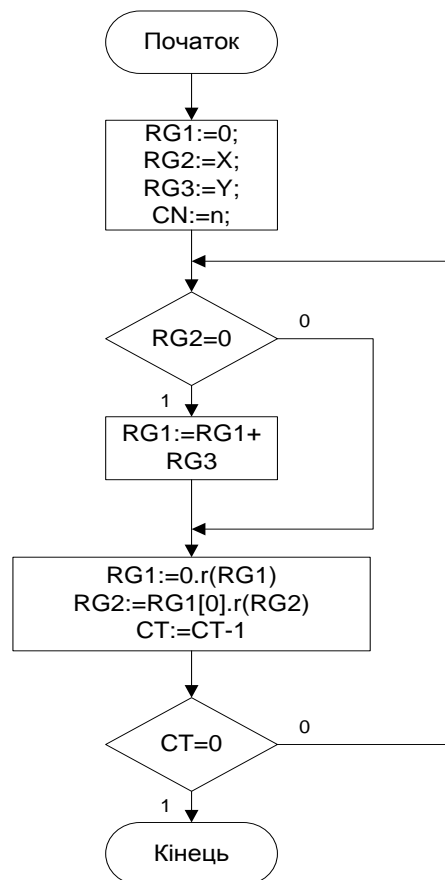


Рис. 1.2 – Мікроалгоритм множення I способом

1.1.4

Таблиця станів реєстрів:

Табл. 2.1 – Таблиця станів реєстрів для множення I способом

№	RG1	RG2	RG3	СТ
ПС	0000000000000000	111110110001111	111110111100001	1111
1	+111110110001111	111111011000111		-1 1110
	011111011000111			
	0011111011000111			
2	+111110110001111	011111101100011		-1 1101
	1011110001010110			
	0101111000101011			
3	+111110110001111	001111110110001		-1 1100
	1101101110111010			
	0110110111011101			
4	+111110110001111	000111111011000		-1 1011
	1101101110111010			
	0111010110110110			
5	0011101011011011	000011111101100		-1 1010
6	0001110101101101	1000011111110110		-1 1001
7	0000111010110110	1100001111111011		-1 1000
8	+111110110001111	111000011111101		-1 0111
	1000110001000101			
	0100011000100010			
9	+111110110001111	111100001111110		-1 0110
	1100001110110001			
	0110000111011000			
10	0011000011101100	011110000111111		0101
11	+111110110001111	101111000011111		-1 0100
	1010111001111011			
	0101011100111101			
12	+111110110001111	010111100001111		-1 0011
	1101010011001100			
	0110101001100110			
13	+111110110001111	101011110000111		-1 0010
	1110011111110101			
	0111001111111010			
14	+111110110001111	110101111000011		-1 0001
	1111000110001001			
	0111100011000100			
15	+111110110001111	111010111100001		-1 0000
	1111011001010011			
	0111101100101001			

1.1.5 Функціональна схема:

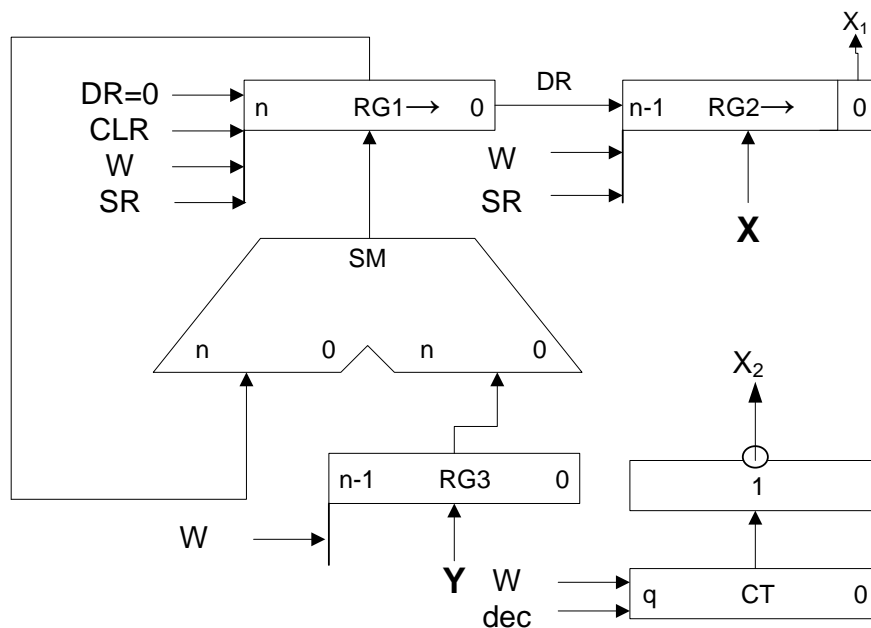


Рис. 1.3 – Функціональна схема з відображенням управляючих сигналів

1.1.6 Закодований мікроалгоритм:

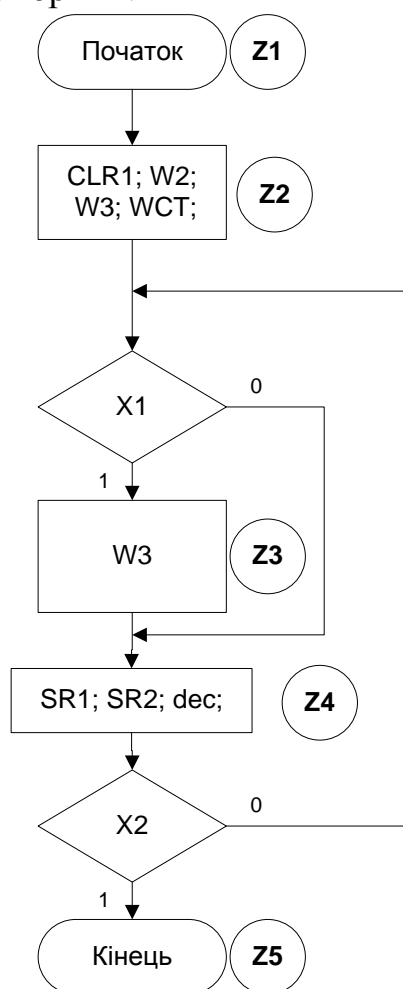


Рис. 1.4 – Закодований мікроалгоритм множення I способом

1.1.7 Граф управляющего автомата Мура:

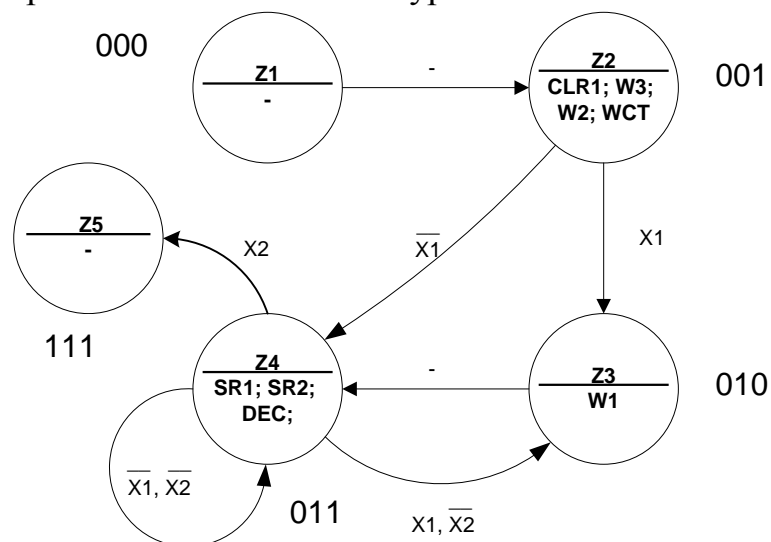


Рис. 1.5 – Граф автомата на I спосіб множення

1.1.8 Обробка порядку результату:

$$P = P_x + P_y = 1000 + 101 = 1101$$

1.1.9 Запис результату в пам'ять

3H	M														
1	1	1	1	1	0	1	1	0	0	1	0	1	0	0	1

Зн	Р						
0	0	0	0	1	1	0	1

1.2 Другий спосіб множення:

1.2.1 Вираз $Z = XY = Yx_12^{-1} + Yx_22^{-2} + \dots + Yx_i2^{-i} + \dots + Yx_n2^{-n}$ можна подати у вигляді:

$$Z = (((\dots((0 + Y2^{-n}x_n) + Y2^{-n+1}x_{n-1}) + \dots + Y2^{-1}x_1$$

Очевидно, що процес множення може бути зведений до n -кратного виконання циклу:

$$Z_i = Z_{i-1} + Y_i x_{n-i+1}, Y_i = 2Y_{i-1}, \text{ з початковими значеннями } i = 1, Z_0 = 0, Y_0 = Y2^{-n}.$$

У розглянутому способі множення здійснюється з молодших розрядів, множене зсувається вліво, а сума часткових добутків залишається нерухомою.

1.2.2 Операційна схема:

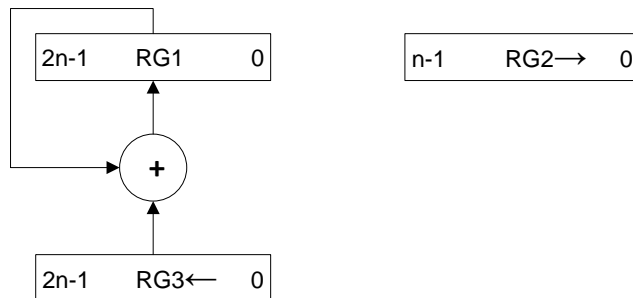


Рис. 1.6 Операційна схема II способу множення

1.2.3 Змістовний мікроалгоритм:

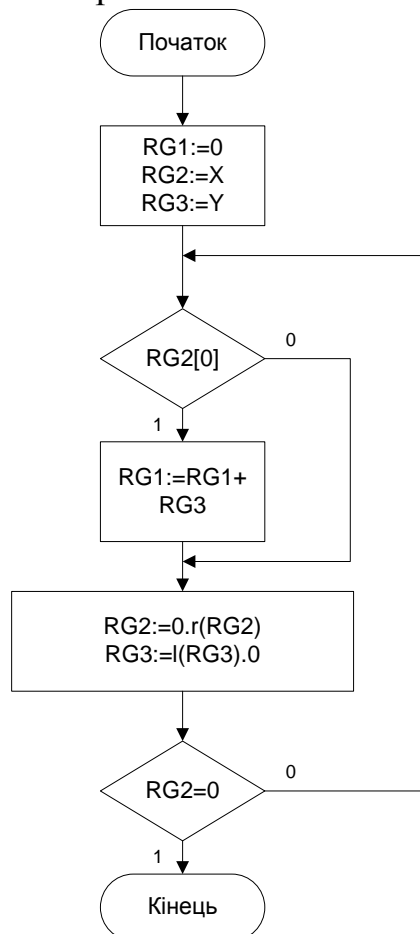


Рис. 1.7 Мікроалгоритм множення II способом

1.2.4 Таблиця станів регістрів:

Табл. 1.2 – Таблиця станів регістрів при множенні II способом

№	RG1	RG3	RG2
ПС	00000000000000000000000000000000	0000000000000000111110110001111	111110110001111
1	+0000000000000000111110110001111 0000000000000000111110110001111	00000000000000001111101100011110	011111011000111
2	+00000000000000001111101100011110 0000000000000010111100010101101	000000000000000011111011000111100	001111101100011
3	+000000000000000011111011000111100 0000000000000110110111011101001	0000000000000000111110110001111000	000111110110001
4	+0000000000000000111110110001111000 00000000000001110101101101100001	00000000000000001111101100011110000	000011111011000
5		00000000000011111011000111100000	000001111101100
6		00000000000111110110001111000000	000000111110110
7		00000000011111101100011110000000	000000011111011
8	+000000011111101100011110000000 000000010001100010001011100001	000000011111011000111100000000	000000001111101
9	+000000111111011000111100000000 000000110000111011000111100001	0000000111110110001111000000000	0000000000111110
10		0000011111101100011110000000000	000000000011111
11	+000011111101100011110000000000 000010101110011110110111100001	0000111111011000111100000000000	0000000000001111
12	+000111111011000111100000000000 000110101001100110010111100001	0001111110110001111000000000000	0000000000000111
13	+001111110110001111000000000000 001110011111110101010111100001	0011111101100011110000000000000	0000000000000011
14	+011111011000111100000000000000 011110001100010011010111100001	0111110110001111000000000000000	0000000000000001
15	+111110110001111000000000000000 111101100101001111010111100001	1111101100011110000000000000000	0000000000000000

1.2.5 Функціональна схема:

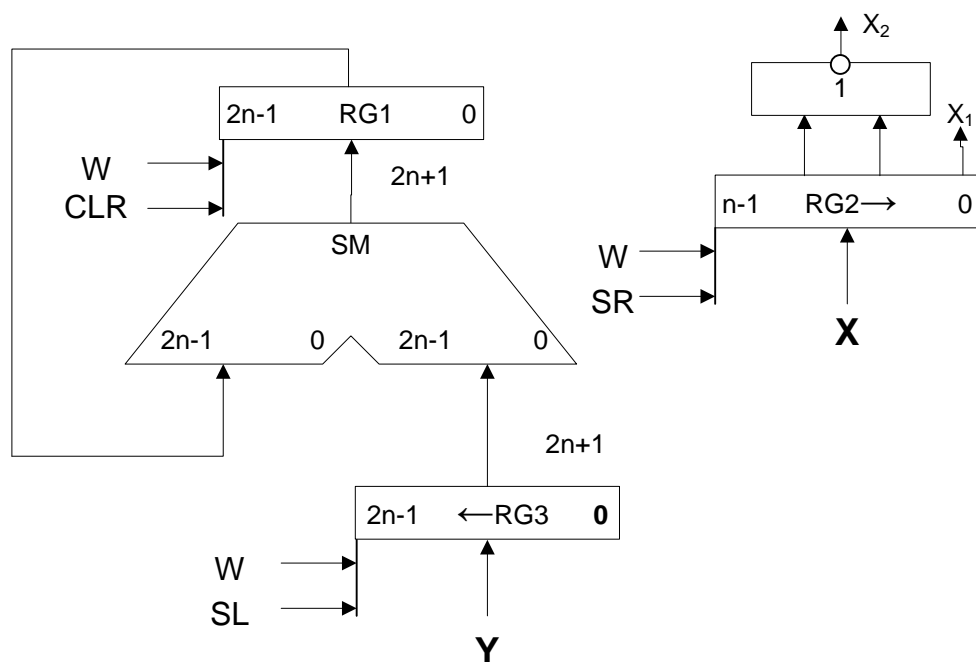


Рис. 1.8 – Функціональна схема з відображенням управляючих сигналів

1.2.6 Закодований мікроалгоритм:

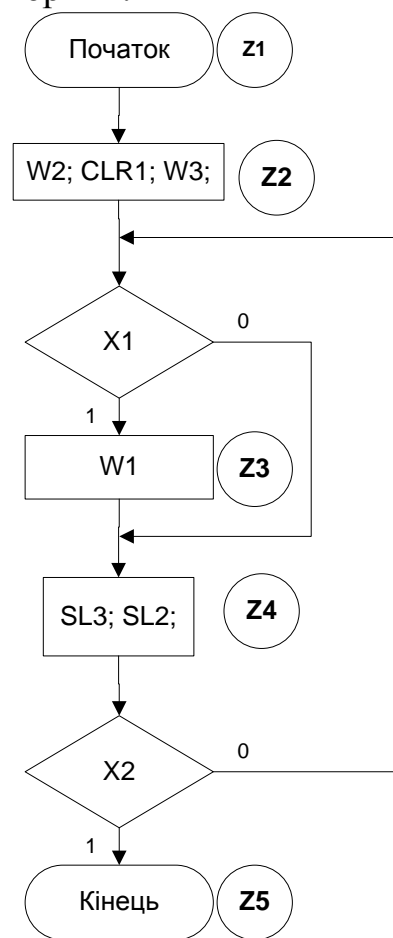


Рис. 1.9 – Закодований мікроалгоритм для множення II способом

1.2.7 Граф управляючого автомата Мура:

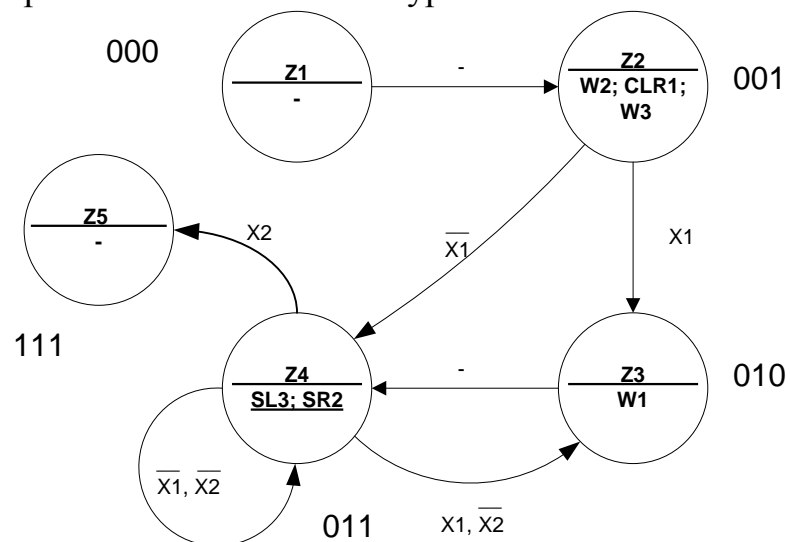


Рис. 1.10 – Граф автомата регулювання управляючих сигналів

1.2.8 Обробка порядку результату:

$$P = P_x + P_y = 1000 + 101 = 1101$$

1.2.9 Запис результату в пам'ять

Зн	М														
1	1	1	1	1	0	1	1	0	0	1	0	1	0	0	1

Зн	Р						
0	0	0	0	1	1	0	1

1.3 Третій спосіб множення:

1.3.1 Вираз $Z = XY = Yx_12^{-1} + Yx_22^{-2} + \dots + Yx_i2^{-i} + \dots + Yx_n2^{-n}$ можна подати у вигляді:

$$Z = (((((0 + Y2^{-n}x_1)2 + Y2^{-n}x_2)2 + \dots + Y2^{-n}x_i)2 + \dots + Y2^{-n}x_n$$

Суму часткових добутків у i -му циклі можна одержати за виразом:

$$Z_i = 2Z_{i-1} + Y2^{-n}x_i, \text{ з початковими значеннями } i = 1, Z_0 = 0.$$

У розглянутому способі множення здійснюється зі старших розрядів множника, сума часткових добутків зсувається вліво, а множене нерухоме.

1.3.2 Операційна схема:

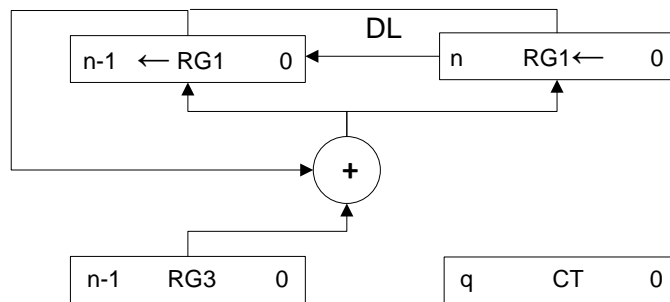


Рис. 1.11 – Операційна схема пристрою для множення III способом

1.3.3 Змістовний мікроалгоритм:

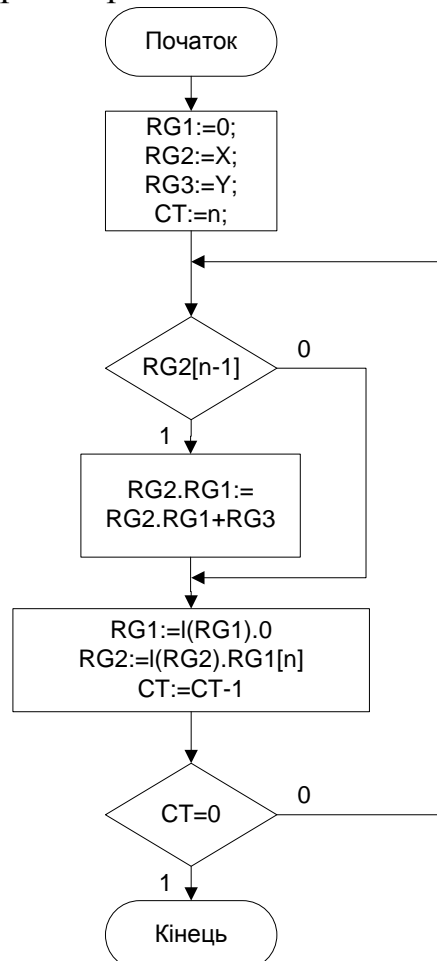


Рис. 1.12 – Мікроалгоритм множення III способом

1.3.4

Таблиця станів регістрів:

Табл. 1.3 Таблиця станів регістрів при множенні ІІІ способом

№	RG2	RG1	RG3	СТ
ПС	111110110001111	0000000000000000	111110110001111	1111
1		+111110110001111		1110
	111110110001111	0111110110001111		
	111101100011110	1111101100011110		
2		+111110110001111		1101
	111101100011111	0111100010101101		
	111011000111110	1111000101011010		
3		+111110110001111		1100
	111011000111111	0110111011101001		
	110110001111110	1101110111010010		
4		+111110110001111		1011
	110110001111111	0101101101100001		
	101100011111110	1011011011000010		
5		+111110110001111		1010
	101100011111111	0011010001010001		
	011000111111110	0110100010100010		
6	110001111111100	1101000101000100		1001
7		+111110110001111		1000
	110001111111101	0100111011010011		
	100011111111010	1001110110100110		
8		+111110110001111		0111
	100011111111011	0001101100110101		
	000111111111010	0011011001101010		
9	001111111101100	0110110011010100		0110
10	011111111101100	1101100110101000		0101
11	111111110110001	1011001101010000		0100
12		+111110110001111		0011
	111111110110010	0011000011011111		
	111111101100100	0110000110111110		
13		+111110110001111		0010
	111111101100100	1101111101001101		
	111111011001001	1011111010011010		
14		+111110110001111		0001
	111111011001010	0011110000101001		
	111110110010100	0111100001010010		
15		+111110110001111		0000
	111110110010100	1111010111100001		
	111101100101001	1110101111000010		

1.3.5 Функціональна схема:

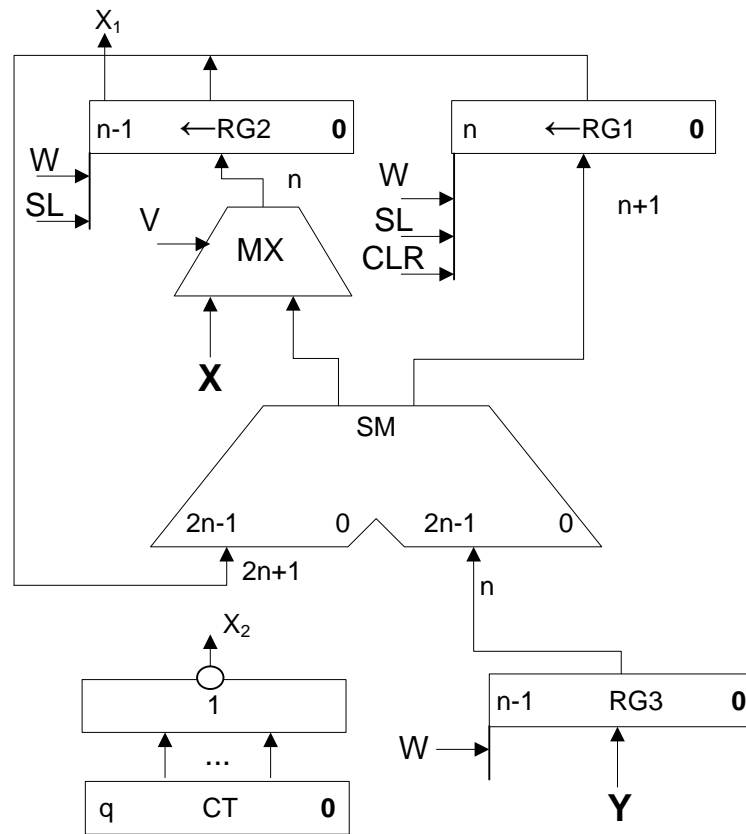


Рис. 1.13 Функціональна схема множення III способом

1.3.6 Закодований мікроалгоритм:

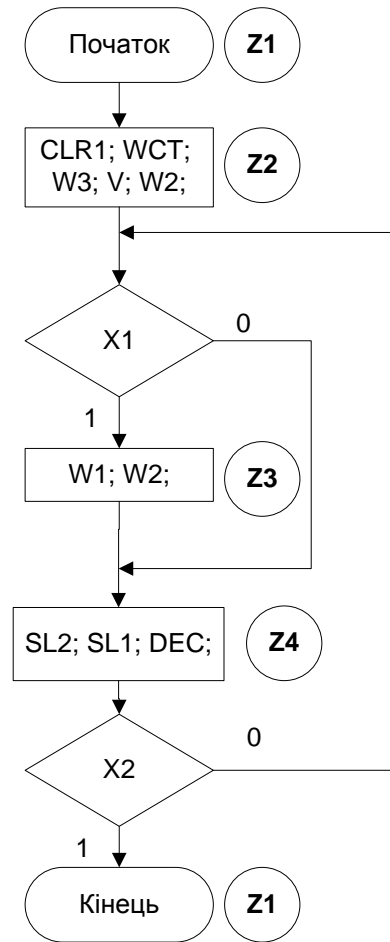


Рис. 1.14 – Закодований мікроалгоритм для множення III способом

1.3.7 Граф управляючого автомата Мура:

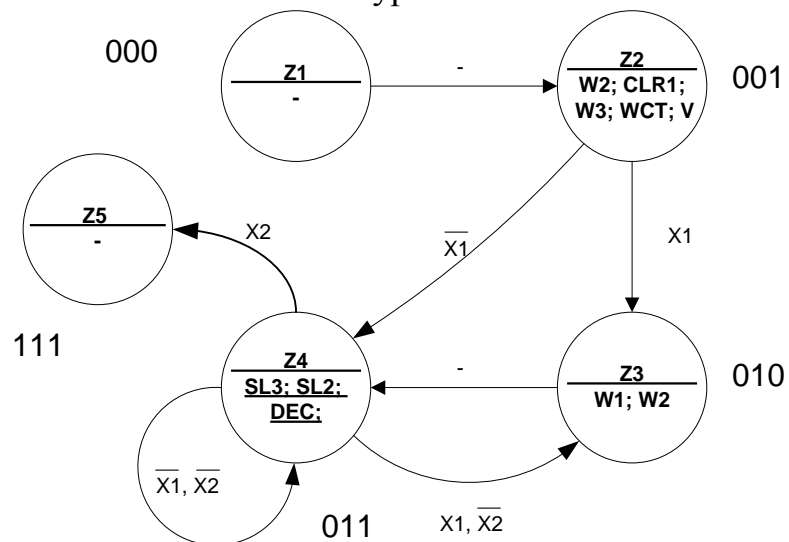


Рис. 1.15 – Граф управляючого автомата

1.3.8 Обработка порядка результата:

$$P = P_x + P_y = 1000 + 101 = 1101$$

1.3.9 Запись результата в память

Зн	М														
1	1	1	1	1	0	1	1	0	0	1	0	1	0	0	1

Зн	Р						
0	0	0	0	1	1	0	1

1.4 Множення IV-им способом:

1.4.1 Вираз $Z = XY = Yx_12^{-1} + Yx_22^{-2} + \dots + Yx_i2^{-i} + \dots + Yx_n2^{-n}$ можна подати у вигляді:

$$Z = (((((0 + Y2^{-1}x_1) + Y2^{-2}x_2) + \dots + Y2^{-i}x_i) + \dots + Y2^{-n}x_n$$

У цьому випадку процес множення може бути зведений до n-кратного виконання циклу

$$Z_i = Z_{i-1} + Y_{i-1}x_i, Y_i = Y_{i-1}2^{-1}, \text{ з початковими значеннями } i = 1, Y_0 = Y2^{-1}, Z_0 = 0.$$

У розглянутому способі множення здійснюється зі старших розрядів множника, сума часткових добутоків залишається нерухомою, а множене зсувається вправо.

1.4.2 Операційна схема:

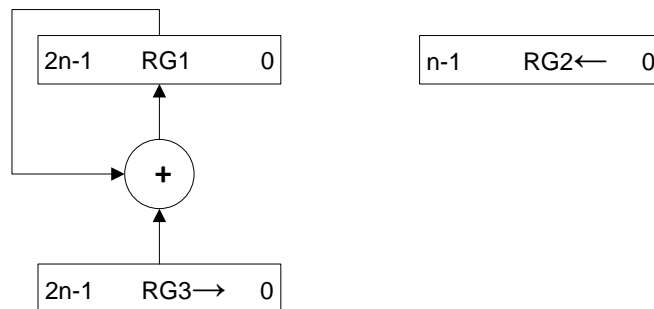


Рис. 1.16 – Операційна схема пристрою множення IV способом

1.4.3 Змістовний мікроалгоритм:

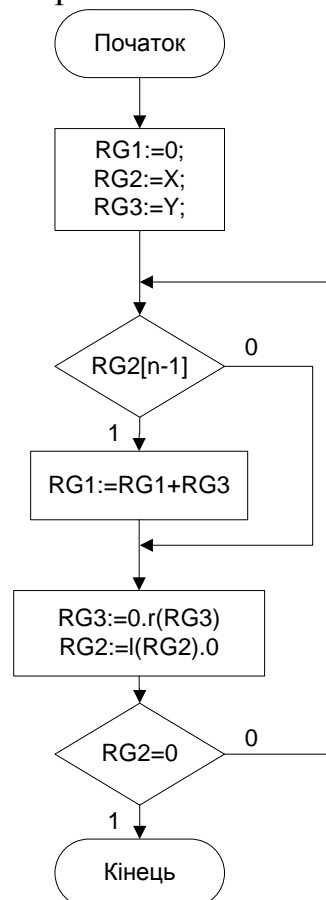


Рис. 1.17 – Мікроалгоритм множення IV способом

1.4.4 Таблиця станів регістрів:

Табл. 1.4 – Таблиця станів регістрів для множення IV способом

№	RG1	RG3	RG2
ПС	00000000000000000000000000000000	11111011000111100000000000000000	111110110001111
1	+11111011000111100000000000000000 01111101100011110000000000000000	01111101100011110000000000000000	111101100011110
2	+01111101100011110000000000000000 10111100010101101000000000000000	00111110110001111000000000000000	111011000111100
3	+00111110110001111000000000000000 11011011101110100100000000000000	00011111011000111100000000000000	110110001111000
4	+00011111011000111100000000000000 11101011011011000010000000000000	00001111101100011110000000000000	101100011110000
5	+00000111110111100100000000000000 11110011010001010001000000000000	00000111110110001111000000000000	011000111100000
6		00000011111011000111100000000000	110001111000000
7	+00000011111011000111100000000000 11110101001110110100110000000000	00000001111101100011110000000000	100011110000000
8	+00000001111101100011110000000000 11110110001101100110101000000000	00000000111110110001111000000000	000111100000000
9		00000000011111011000111100000000	
10		00000000001111101100011110000000	
11		00000000000111110110001111000000	111100000000000
12	+000000000001111101100011110000 1111011001000110000110111110000	000000000000111110110001111000	111000000000000
13	+000000000000111110110001111000 1111011001001101111101001101000	000000000000011111011000111100	110000000000000
14	+000000000000011111011000111100 1111011001010001111000010100100	000000000000001111101100011110	100000000000000
15	+000000000000001111101100011110 1111011001010011110101111000010	000000000000000111110110001111	000000000000000

1.4.5 Функціональна схема:

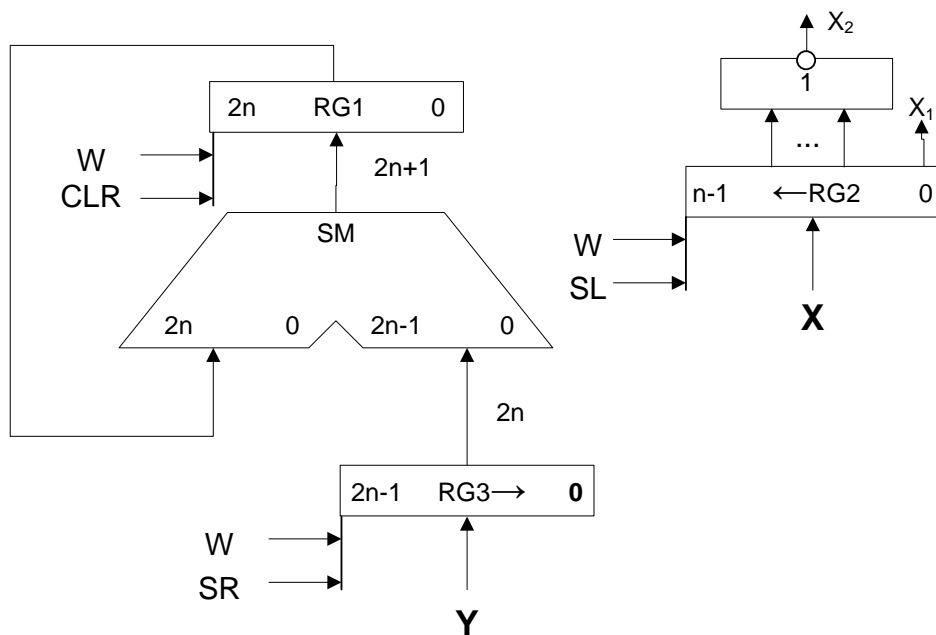


Рис. 1.18 Функціональна схема з відображенням управляючих сигналів

1.4.6 Закодований мікроалгоритм:

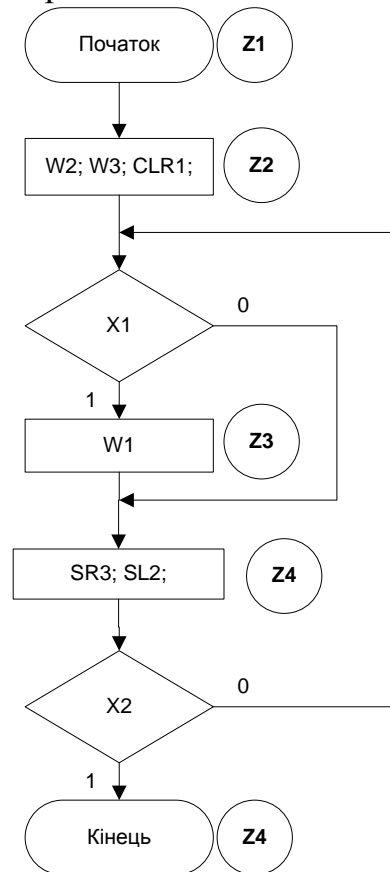


Рис. 1.19 – Закодований мікроалгоритм для множення IV способом

1.4.7 Граф управляючого автомата Мура:

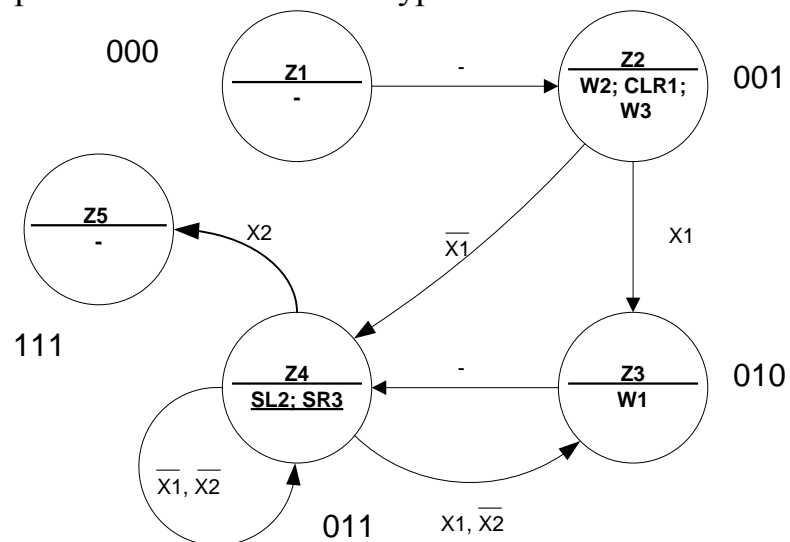


Рис. 1.20 – Граф автомата мура для множення IV способом

1.4.8 Обработка порядка результата:

$$P = P_x + P_y = 1000 + 101 = 1101$$

1.4.9 Запись результата в память

Зн	М														
1	1	1	1	1	0	1	1	0	0	1	0	1	0	0	1

Зн	Р						
0	0	0	0	1	1	0	1

2 Операція ділення чисел

2.1 Ділення першим способом

2.1.1 Під час ділення за першим способом 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 записане ділене X). Ділене Y знаходиться в регістрі RG1. Результат формується в регістрі RG3 за (n+1) циклів. Алгоритм ділення зводиться до виконання наступних дій:

- 1) Одержати різницю $R_0 = X - Y$. Якщо $R_0 \geq 0$, то цифра частки Z_0 , що має вагу 2^0 , дорівнює 1, а за $R_0 < 0$ – дорівнює 0. Різниця R_0 є залишком.
- 2) Подвоїти залишок (тобто одержати значення $2R_i$).
- 3) За $2R_i < 0$ додати Y, в зворотному випадку, якщо $2R_i \geq 0$, відняти Y. Якщо знову отриманий залишок $R_{i+1} \geq 0$, то $Z_{i+1} = 1$, інакше $Z_{i+1} = 0$.
- 4) Повторити дії, описані в пунктах 2 та 3, (n-1) раз.

2.1.2 Операційна схема:

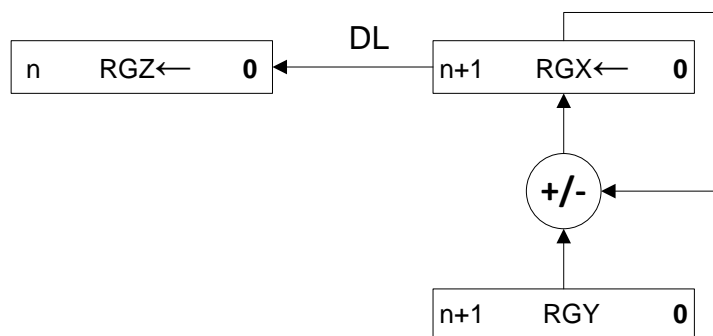


Рис. 2.1 – Операційна схема ділення I способом

2.1.3 Функціональний мікроалгоритм:

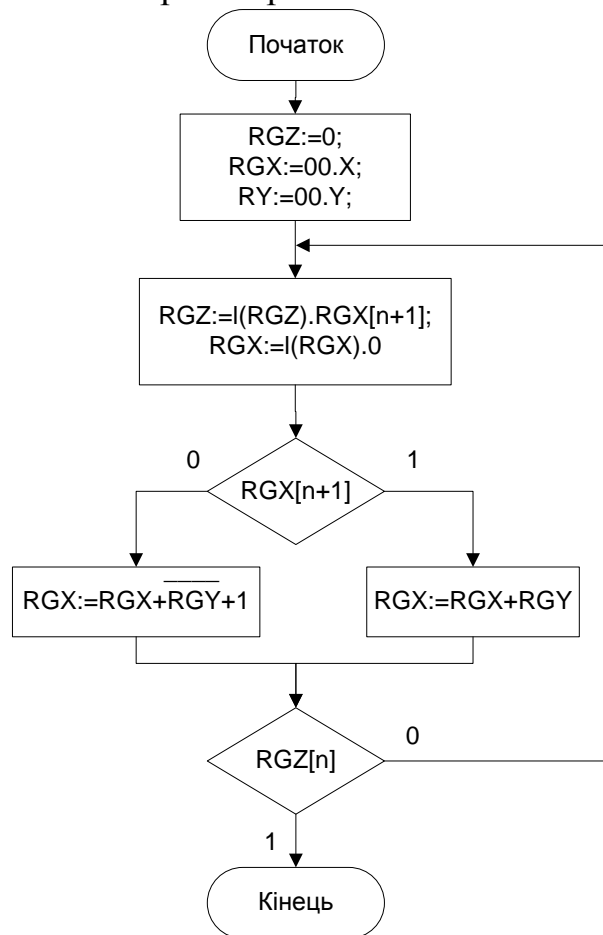


Рис. 2.2 Мікроалгоритм ділення I способом

2.1.4 Таблиця станів регістрів:

Табл. 2.1 – Таблиця станів регістрів при діленні I способом

№	RGZ	RGX	RGY
ПС	0000000000000000	00111110110001111	00111110111100100
1	0000000000000001	01111101100011110 +11000001001110001 00111110110001111	
2	0000000000000011	01111101100011110 +11000001001110001 00111110110001111	
3	0000000000000111	01111101100011110 +11000001001110001 00111110110001111	
4	0000000000011111	01111101100011110 +11000001001110001 00111110110001111	
5	0000000000111111	01111101100011110 +11000001001110001 00111110110001111	
6	0000000001111111	01111101100011110 +11000001001110001 00111110110001111	
7	0000000011111111	01111101100011110 +11000001001110001	

		00111110110001111	
8	0000000111111111	01111101100011110 +11000001001110001 00111110110001111	
9	0000001111111111	01111101100011110 +11000001001110001 00111110110001111	
10	0000011111111111	01111101100011110 +11000001001110001 00111110110001111	
11	0000111111111111	01111101100011110 +11000001001110001 00111110110001111	
12	0001111111111111	01111101100011110 +11000001001110001 00111110110001111	
13	0011111111111111	01111101100011110 +11000001001110001 00111110110001111	
14	0111111111111111	01111101100011110 +11000001001110001 00111110110001111	
15	1111111111111111	01111101100011110 +11000001001110001 00111110110001111	

2.1.5 Функціональна схема:

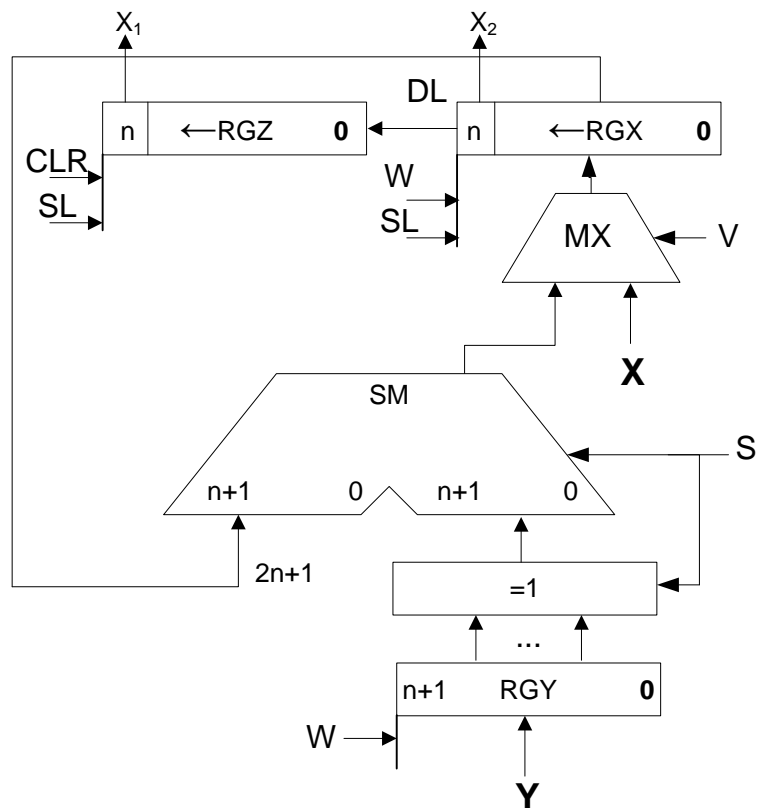


Рис. 2.3 Функціональна схема для ділення I способом

2.1.6 Закодований мікроалгоритм:

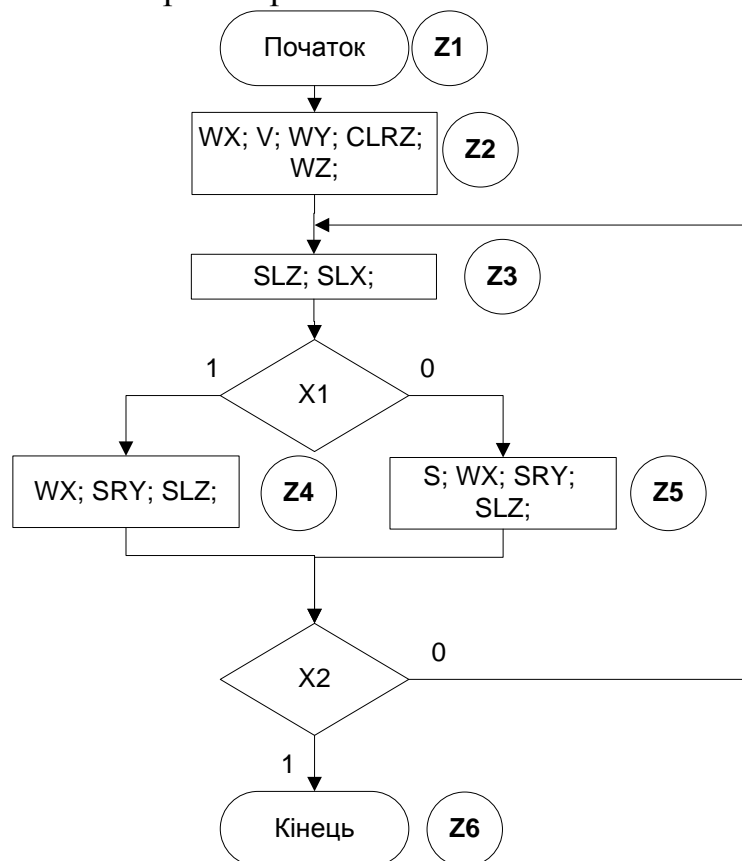


Рис. 2.4 – Закодований мікроалгоритм для ділення I способом

2.1.7 Граф управляючого автомата Мура:

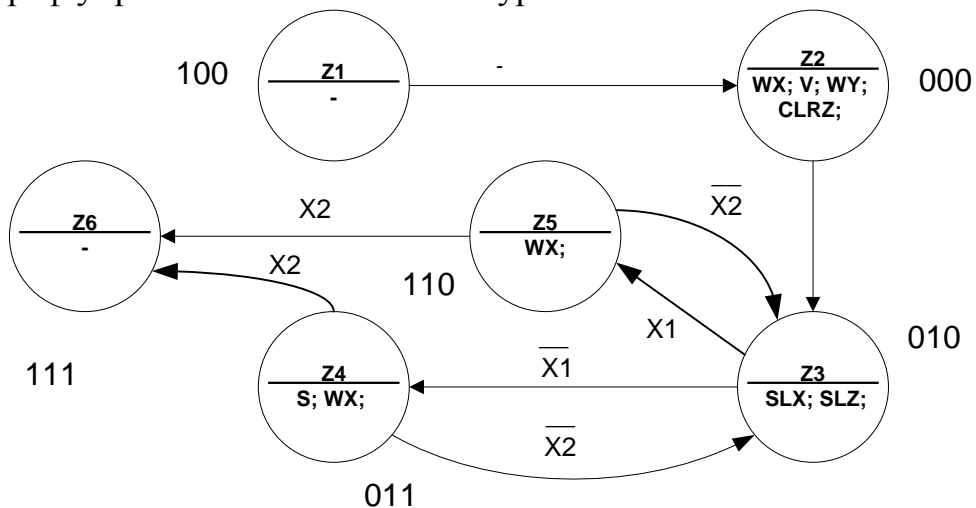


Рис. 2.5 Граф автомата для ділення I способом

2.1.8 Обробка порядку результату:

$$P = P_y - P_x = 1000 - 101 = 11$$

2.1.9 Запис результату $Z=X/Y$ в пам'ять:

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

2.2 Ділення II-им способом

2.2.1 Під час ділення другим способом, який називається діленням із зсувом дільника, збільшується розрядність регістрів RG1, RG3 і суматора SM. В цьому випадку процеси додавання-віднімання і зсуву можуть бути сполучені в часі. Цифра результату формується на виході суматора SM(p).

Алгоритм ділення зводиться до виконання наступних дій:

- 1) Одержати різницю $R_0 = X - Y$. Якщо $R_0 \geq 0$, то цифра частки Z_0 , що має вагу 2^0 , дорівнює 1, а за $R_0 < 0$ – дорівнює 0. Різниця R_0 є залишком.
- 2) Зменшити вдвічі дільник.
- 3) За $R_i < 0$ додати Y , в зворотному випадку, якщо $R_i \geq 0$, відняти Y . Якщо знову отриманий залишок $R_{i+1} \geq 0$, то $Z_{i+1} = 1$, інакше $Z_{i+1} = 0$.
- 4) Повторити дії, описані в пунктах 2 та 3, (n-1) раз.

2.2.2 Операційна схема:

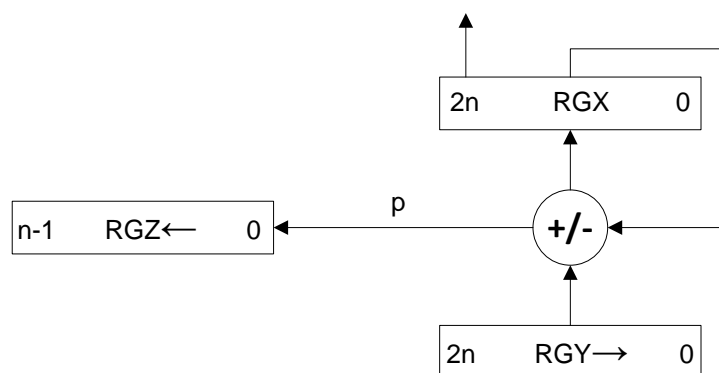


Рис. 2.6 Операційна схема ділення II способом

2.2.3 Функціональний мікроалгоритм:

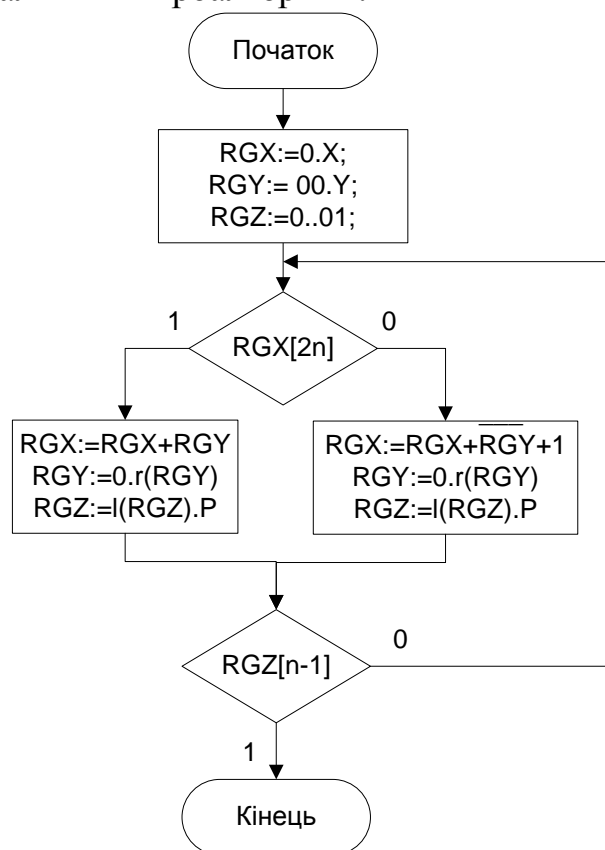


Рис. 2.7 – Мікроалгоритм ділення II способом

2.2.4 Таблиця станів регістрів:

Табл. 2.2 Таблиця станів регістрів ділення II способом

№	RGZ	RGX	RGY
ПС	0000000000000001	011111011000111100000000000000	001111101100011110000000000000
1	0000000000000011	+110000010011100010000000000000 001111101100011110000000000000	000111110110001111000000000000
2	0000000000000111	+111000001001110001000000000000 000111110110001111000000000000	000011111011000111100000000000
3	0000000000001111	+111100000100111000100000000000 000011111011000111100000000000	000001111101100011110000000000
4	000000000011111	+111110000010011100010000000000 000001111101100011110000000000	000000111110110001111000000000
5	000000000111111	+111111000001001110001000000000 000000111110110001111000000000	000000011111011000111100000000
6	000000001111111	+111111100000100111000100000000 000000011111011000111100000000	000000001111101100011110000000
7	000000011111111	+111111110000010011100010000000 000000001111101100011110000000	000000000111110110001111000000
8	000000111111111	+111111111000001001110001000000 000000000111110110001111000000	000000000011111011000111100000
9	000001111111111	+111111111100000100111000100000 000000000011111011000111100000	000000000001111101100011110000
10	000011111111111	+111111111110000010011100010000 000000000001111101100011110000	000000000000111110110001111000
11		+111111111111000001001110001000	

	0001111111111111	0000000000000111110110001111000	0000000000000011111011000111100
12	0011111111111111	+111111111111100000100111000100 0000000000000011111011000111100	0000000000000001111101100011110
13	0111111111111111	+11111111111110000010011100010 000000000000001111101100011110	0000000000000000111110110001111
14	1111111111111111	+1111111111111000001001110001 000000000000000111110110001111	0000000000000000011111011000111

2.2.5 Функціональна схема:

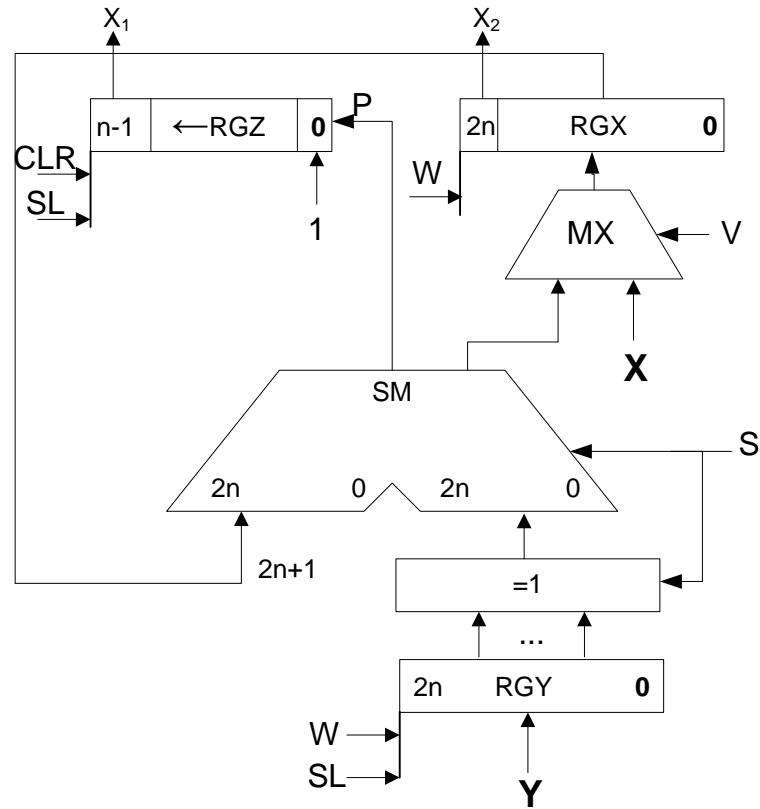


Рис. 2.8 – Функціональна схема з відображенням управляючих сигналів

2.2.6 Закодований мікроалгоритм:

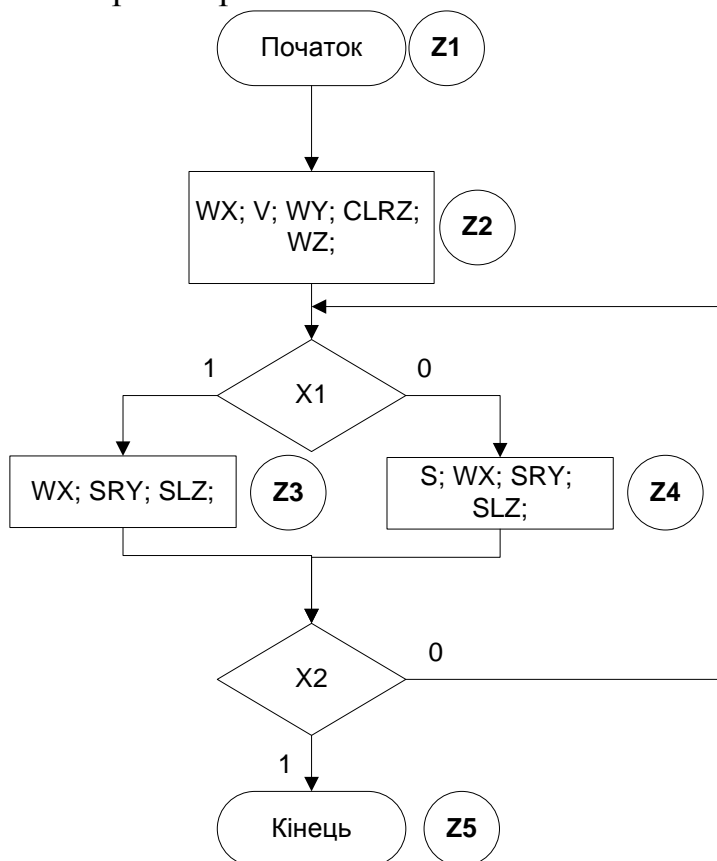


Рис. 2.9 Закодований мікроалгоритм для ділення II способом

2.2.7 Граф управляючого автомата Мура:

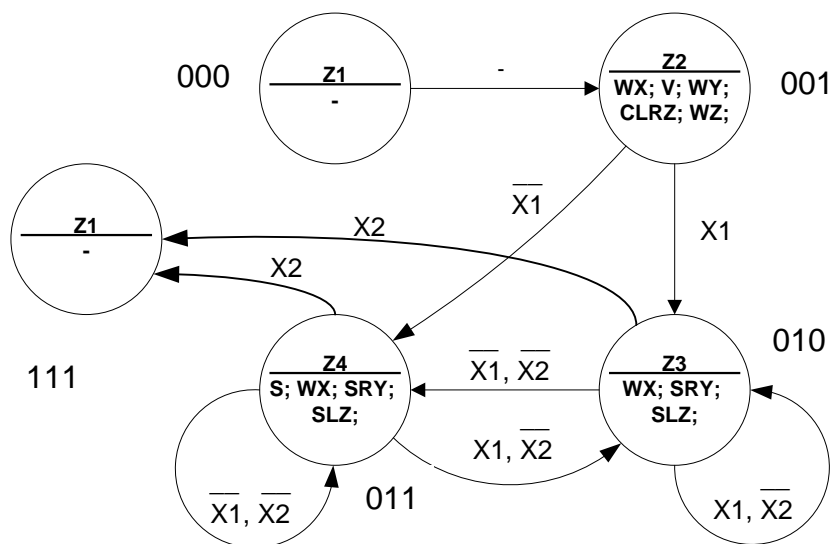


Рис. 2.10 Граф управляючого автомата для ділення II способом

2.2.8 Обробка порядку результату:

$$P = P_y - P_x = 1000 - 101 = 11$$

2.2.9 Запис результату $Z=Y/X$ в пам'ять:

Н	М													

Н	Р						

3 Додавання чисел

1) $P_x > P_y$, отже $P_z = P_x = 0.0001000$;
 $\Lambda = 11$

2) Вирівнюємо порядки

Λ	M_y
11	00.111110110001111
10	00.011111011000111
01	00.001111101100011
00	00.000111110110001

3) Перетворимо в ДК та підсумуванням:

$$\begin{array}{r}
 +P_x = 11.000001001110001 \\
 P_y = 00.000111110110001 \\
 \hline
 P_z = 11.001001000100010
 \end{array}$$

4) Перетворимо результат назад в ПК:

$$P_z = 11.110110111011110$$

Решту 9 пунктів виконаємо для нормалізації:

3.1 В машинах підсумування і віднімання кодів виконується на суматорах.

Віднімання замінюється на додавання за правилом: один операнд змінює знак.

Прямий код не застосовується для підсумування.

В арифметичному пристрої знаковий розряд подвоюється, щоб не було переповнення розрядної сітки.

Перед підсумування необхідно виконати вирівнювання порядків операндів.

Після цього здійснюється підсумування мантис операндів на суматорі за участі суматора переносу.

3.2 Операційна схема:

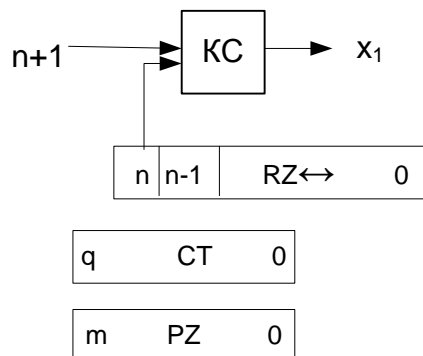


Рис.3.1 – Операційна схема нормалізації при додаванні

3.3 Функціональний мікроалгоритм:

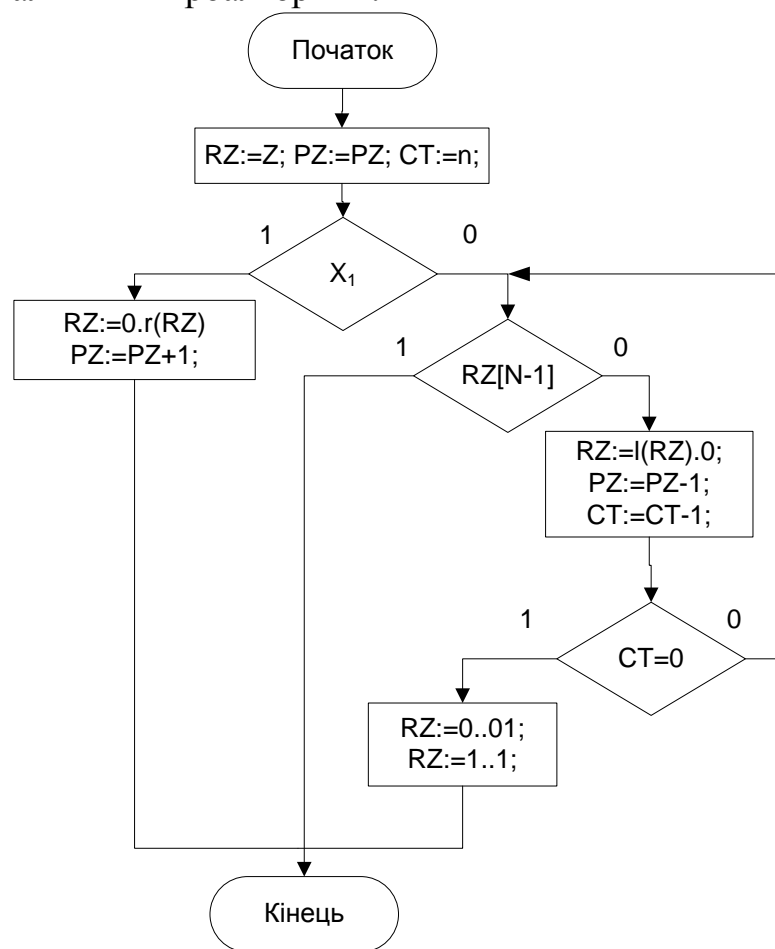


Рис. 3.2 – Мікроалгоритм додавання

3.4 Таблица станів регістрів:

Табл. 3.1 Таблица станів регістрів при додаванні

№	RZ	PZ	CT
ПС	1110110111011110	00001000	1111

3.5 Функціональна схема:

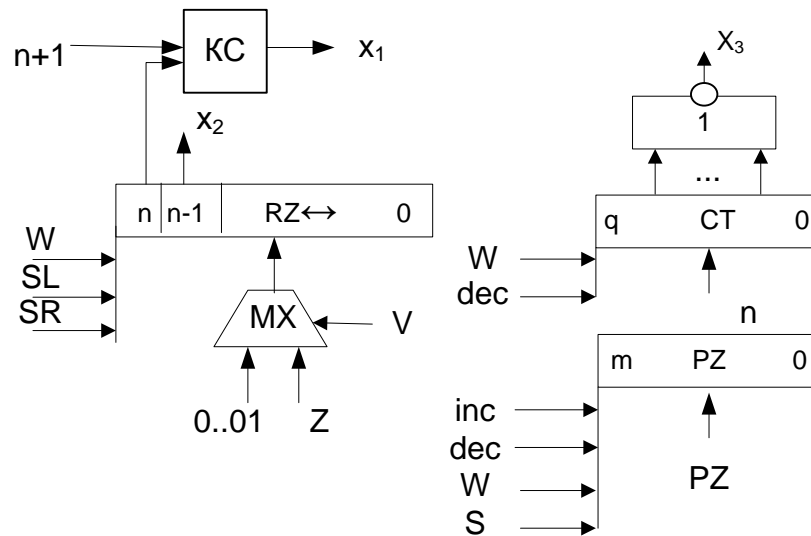


Рис. 3.3 – Функціональна схема з відображенням управляючих сигналів для додавання

3.6 Закодований мікроалгоритм:

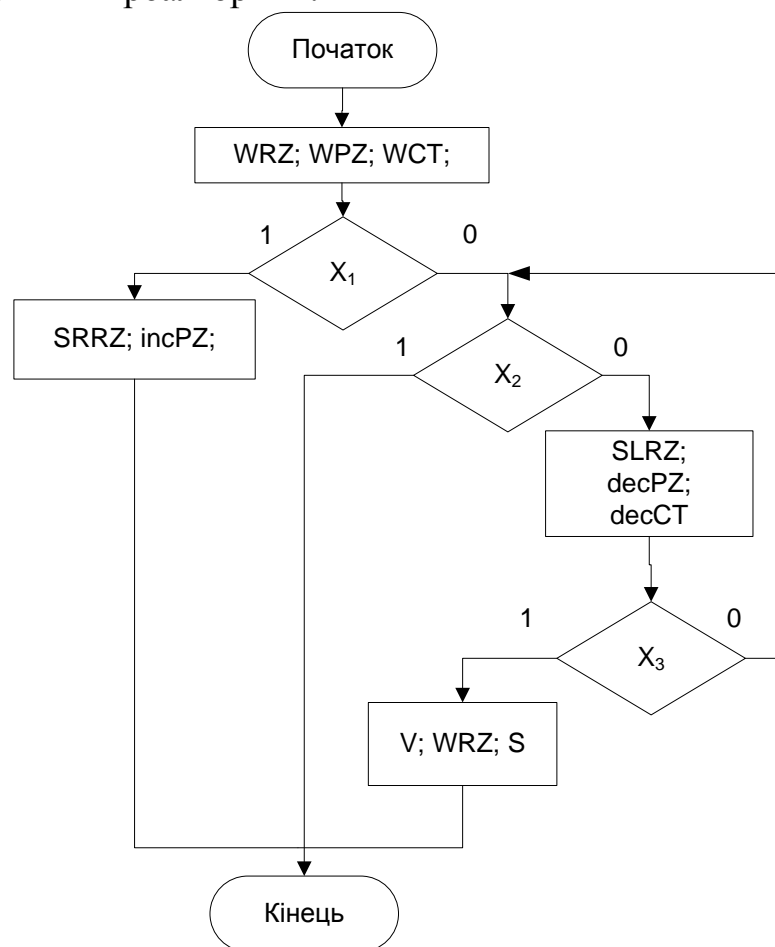


Рис. 3.4 – Закодований мікроалгоритм для додавання

3.7 Граф управляющего автомата Мура:

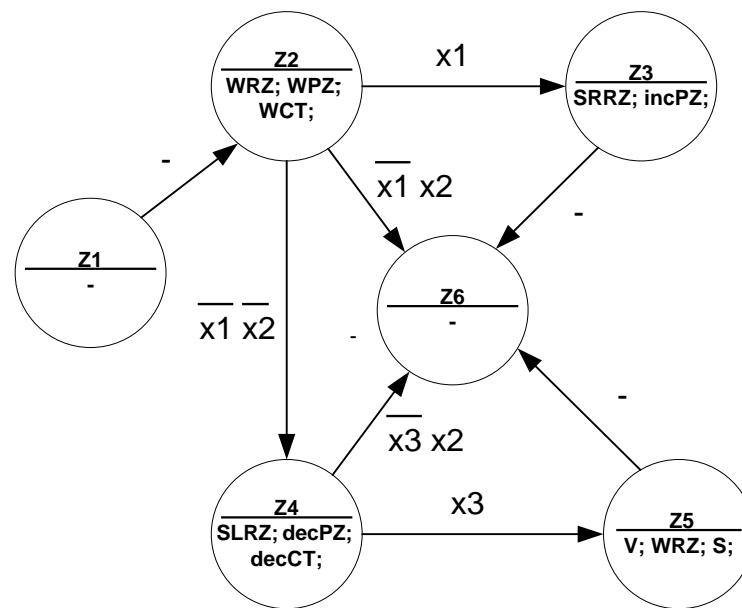


Рис. 3.5 Граф автомата управління пристроєм додавання

3.8 Обробка порядку результату входить в саму схему.

3.9 Запис результату $Z=Y+X$ в пам'ять:

Зн	М														
1	1	1	0	1	1	0	1	1	1	0	1	1	1	1	0

Зн	Р						
0	0	0	0	1	0	0	0

4 Добування кореня

4.1 Розглянемо метод обчислення функції $Y = \sqrt{X}$ за послідовного введення коду аргументу X зі старших розрядів. Такий метод дозволяє виконувати процеси введення аргументу та обчислення функції у режимі суміщення.

Будемо вважати, що значення аргументу подане нормалізованим n -розрядним числом у позиційній системі числення з основою 2.

Процес обчислення квадратного кореня зводиться до послідовного виконання n кроків. Перед початком обчислення $X_0 = Y_0 = R_0 = 0$. На кожному i -му кроці ($i \in (1, n)$) цифру y_i визначають відповідно системі:

$$y_i = \begin{cases} 0, & \text{якщо } R_{i-1} < N \\ 1, & \text{якщо } N_i < R_{i-1} < N_2 \\ 2, & \text{якщо } R_{i-1} \geq N_2 \end{cases}$$

Значення Y_i та R_i – за формулами:

$$Y_i = Y_{i-1} + y_i 2^{-i}$$

$$R_i = 2R_{i-1} + x_i 2^{-1} - Y_{i-1} y_i - y_i^2 2^{-i-1}$$

При цьому формула обчислення Y_i забезпечує автоматичне перетворення коду результату з надлишкової системи числення в не надлишкову, якщо операцію додавання виконувати в не надлишковій системі числення.

Процес обчислення квадратного кореня за запропонованим алгоритмом закінчується безпосередньо після введення останнього розряду коду аргументу.

Це дозволяє зменшити час обчислення в тих випадках, коли швидкість надходження чергових розрядів аргументу визначається зовнішніми стосовно операційного блоку факторами.

4.2 Операційна схема:

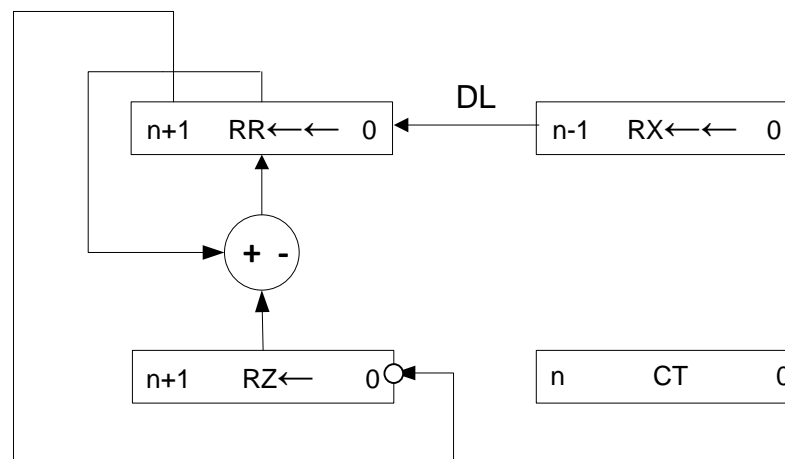


Рис. 4.1 Операційна схема пристрою для добування кореня

4.3 Функціональний мікроалгоритм:

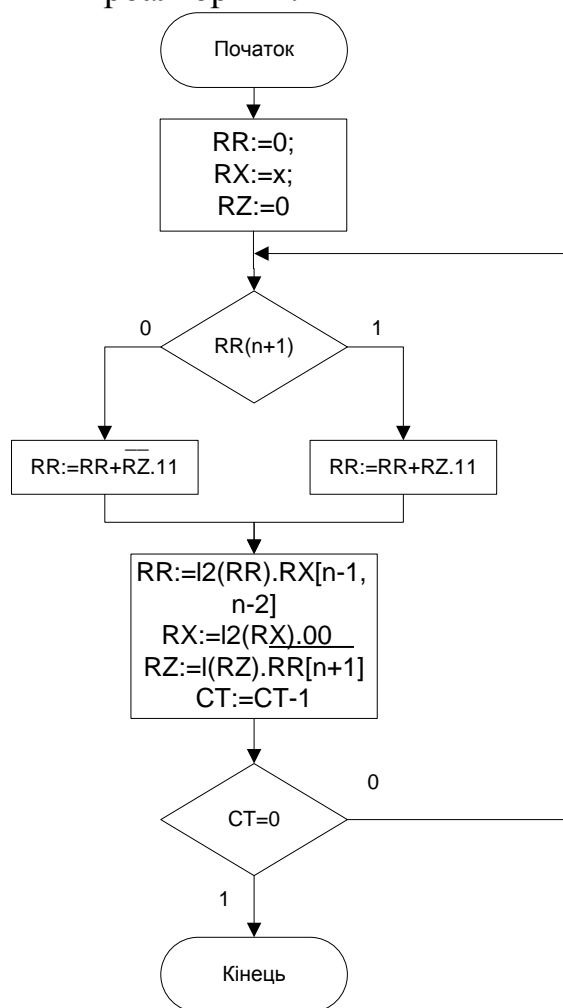


Рис. 4.2 Мікроалгоритм добування кореня

4.4 Таблиця станів регістрів:

Табл. 4.1 Таблиця станів регістрів пристрою для добування кореня

№	СТ	RZ	RR	RX
ПС	1111	0000000000000000	0000000000000000	111110110001111
ПЗ	1111	0000000000000000	0000000000000001	111011000111100
1	1110	0000000000000001	+1111111111111111 00000000000000010 00000000000001011	101100011110000
2	1101	0000000000000011	+11111111111111011 00000000000000110 00000000000011010	110001111000000
3	1100	0000000000000111	+11111111111110011 00000000000001101 00000000000110111	000111100000000
4	1011	0000000000001111	+11111111111100011 00000000000011010 00000000001101000	011110000000000
5	1010	0000000000011111	+11111111111000011 00000000000101011 00000000010101101	111000000000000
6			+111111111110000011	

	1001	0000000001111111	00000000000110000 00000000011000011	1000000000000000
7	1000	0000000001111110	+11111111100000011 1111111111000110 11111111100011010	0000000000000000
8	0111	0000000011111101	+00000000111111011 00000000100010101 00000010001010100	0000000000000000
9	0110	000000111111011	+11111110000001011 00000000001011111 00000000101111100	0000000000000000
10	0101	000001111110110	+11111100000010011 11111100110001111 11110011000111100	0000000000000000
11	0100	000011111101100	+00000111111011011 11111011000010111 11101100001011100	0000000000000000
12	0011	0001111111011000	+00001111110110011 11111100000001111 11110000000111100	0000000000000000
13	0010	0011111110110001	+000111111101100011 00001111110011111 001111111001111100	0000000000000000
14	0001	0111111101100010	+11000000100111011 11111111110110111 11111111011011100	0000000000000000
15	0000	111111011000101	+01111110110001011 01111110001100111 11111000110011100	0000000000000000

4.5 Функціональна схема:

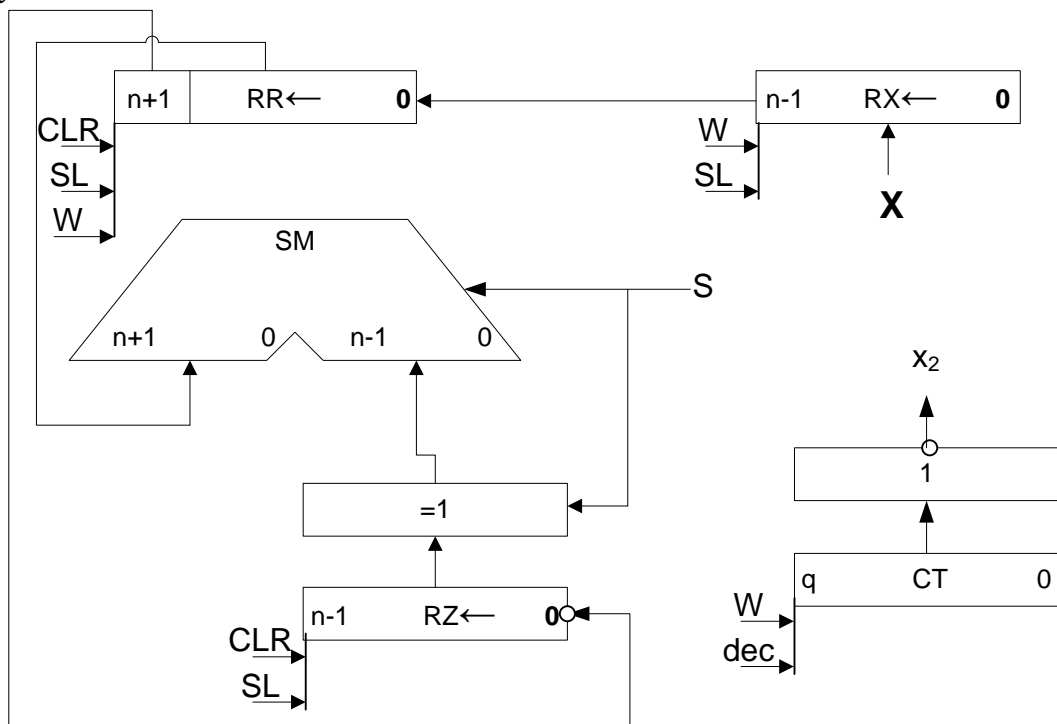


Рис. 4.3 Функціональна схема добування кореня

4.6 Закодований мікроалгоритм:

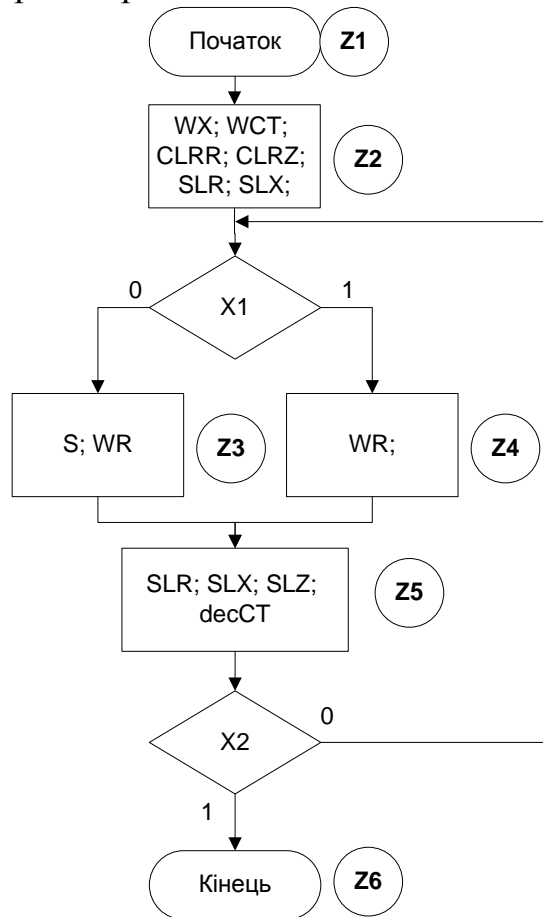


Рис. 4.4 Закодований мікроалгоритм добування кореня.

4.7 Граф управляючого автомата Мура:

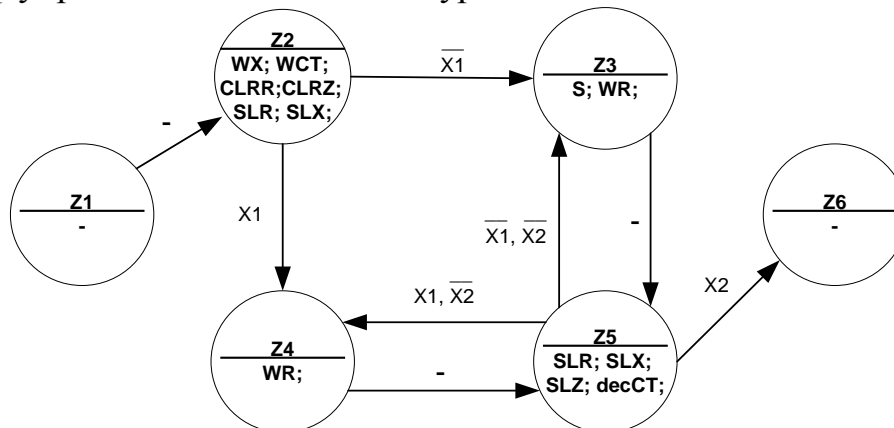


Рис. 4.5 Граф Автомата регулювання пристрою добування кореня

4.8 Обробка порядку результату:

$$P = P_x/2 = 1000/10 = 0100$$

4.9 Запис результату $Z = \sqrt{x}$ в пам'ять:

ЗН	М													
0	1	1	1	1	1	0	0	0	1	1	0	0	1	1

ЗН	Р							
0	0	0	0	0	1	0	0	

5 Синтез управляющего автомата для операционного пристрою добування кореня.

5.1 Таблиця співвідношення управляючих входів операційного автомата і виходів управляючого автомата

Кодування мікрооперацій	
МО	Позначення
WX, WCT, CLRR, CLRZ	Y1
SLR, SLX	Y2
S	Y3
WR	Y4
SLZ, decCT	Y5

5.2 Мікроалгоритм в термінах управляючого автомата

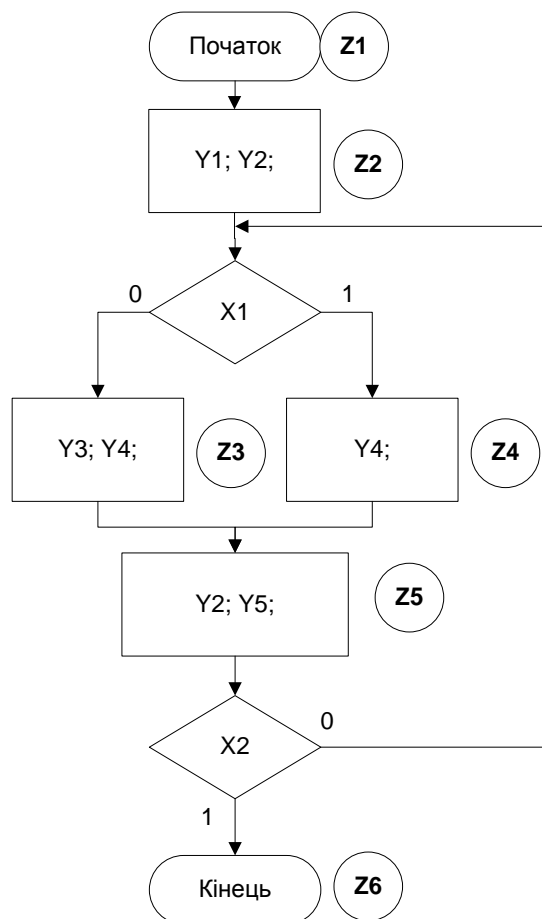


Рис. 5.1 – Закодований алгоритм в термінах управляючого автомата

5.3 Структурна таблиця автомата

Табл. 5.1 – Структурна таблиця автомата

	Старі			Нові			Входи		Виходи					Тригери		
	Q3	Q2	Q1	Q3	Q2	Q1	X1	X2	Y1	Y2	Y3	Y4	Y5	T3	T2	T1
Z1→Z2	0	0	0	0	0	1	-	-	0	0	0	0	0	0	0	1
Z2→Z3	0	0	1	0	1	1	0	-	1	1	0	0	0	0	1	0
Z2→Z4	0	0	1	0	1	0	1	-	1	1	0	0	0	0	1	1
Z3→Z5	0	1	1	1	1	0	-	-	0	0	1	1	0	1	0	1
Z5→Z3	1	1	0	0	1	1	0	0	0	1	0	0	1	1	0	1
Z4→Z5	0	1	0	1	1	0	-	-	0	0	0	1	0	1	0	0
Z5→Z4	1	1	0	0	1	0	1	0	0	1	0	0	1	1	0	0
Z5→Z6	1	1	0	1	1	1	-	1	0	1	0	0	1	0	0	1

5.4 Синтез функцій виходів і переходів

Мінімізація виходів:

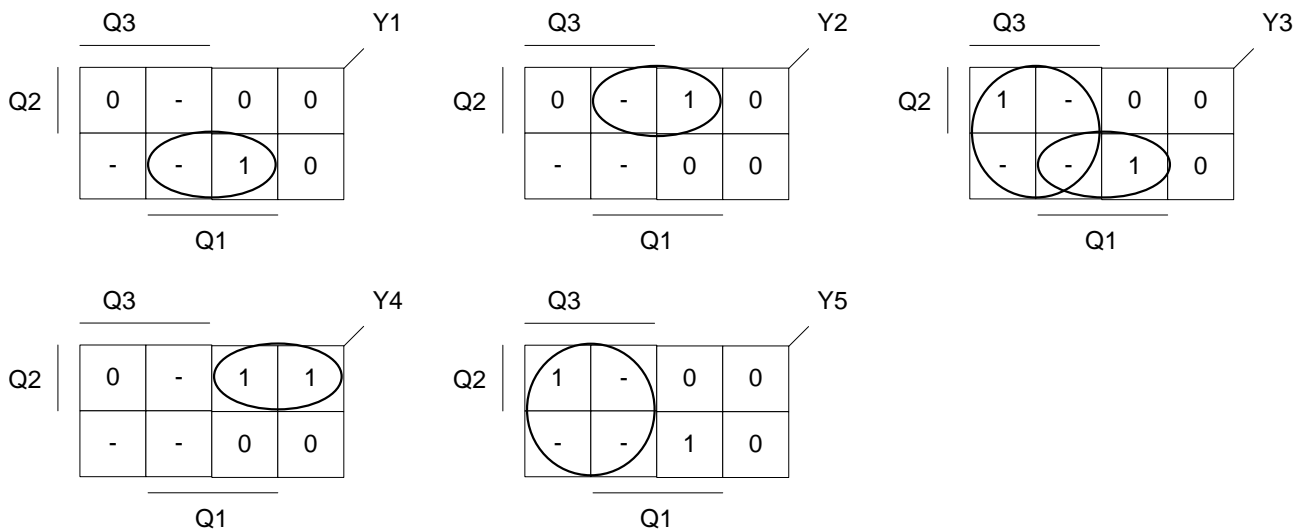


Рис. 5.2 – Мінімізація функцій Y методом діаграм Вейча

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

$$Y_2 = \overline{Q_2}Q_1 \vee Q_3$$

$$Y_3 = Q_2Q_1$$

$$Y_4 = \overline{Q_3}Q_2$$

$$Y_5 = Q_3$$

Мінімізація функцій тригерів:

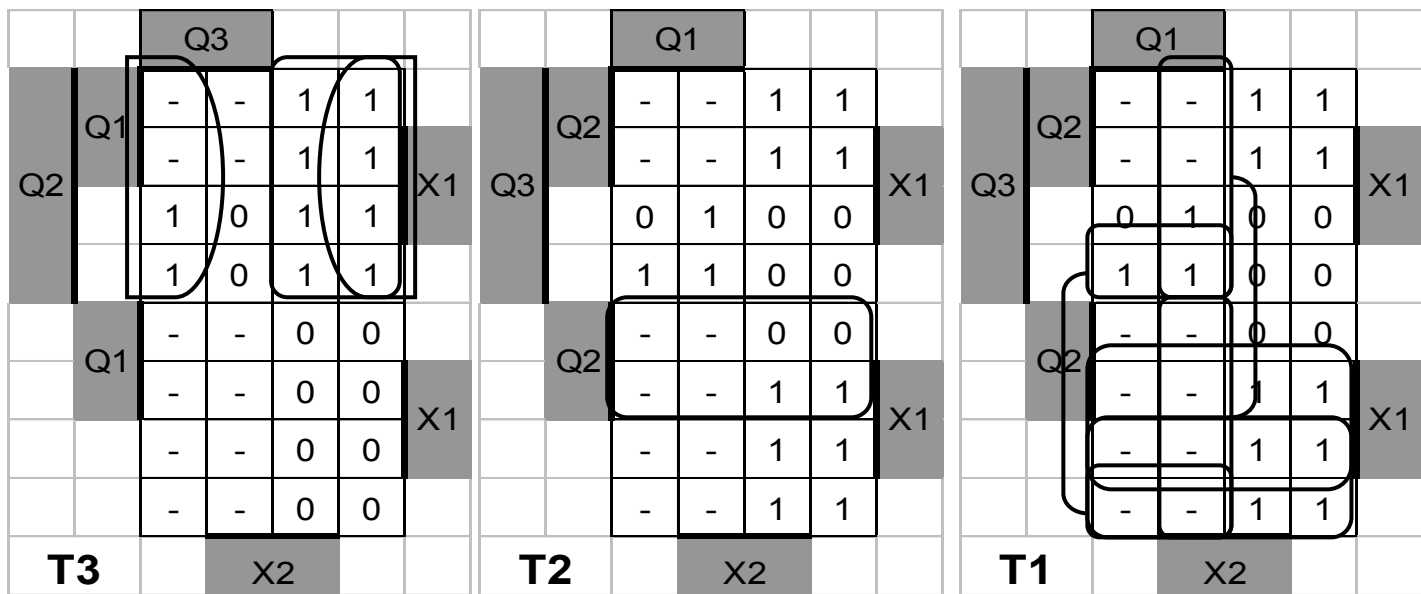


Рис. 5.3 – Діграми Вейча для мінімізації функцій тригерів

$$T_3 = \overline{Q_3}Q_2 \vee \overline{X_2}Q_2$$

$$T_2 = Q_1 \overline{Q_2}$$

$$T_1 = Q_1 Q_2 \vee X_2 Q_3 \vee \overline{X_1} Q_3 \overline{Q_1} \vee \overline{Q_2} \overline{Q_1} \vee X_1 \overline{Q_2}$$

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

$$Y3 = \overline{Q2Q1}$$

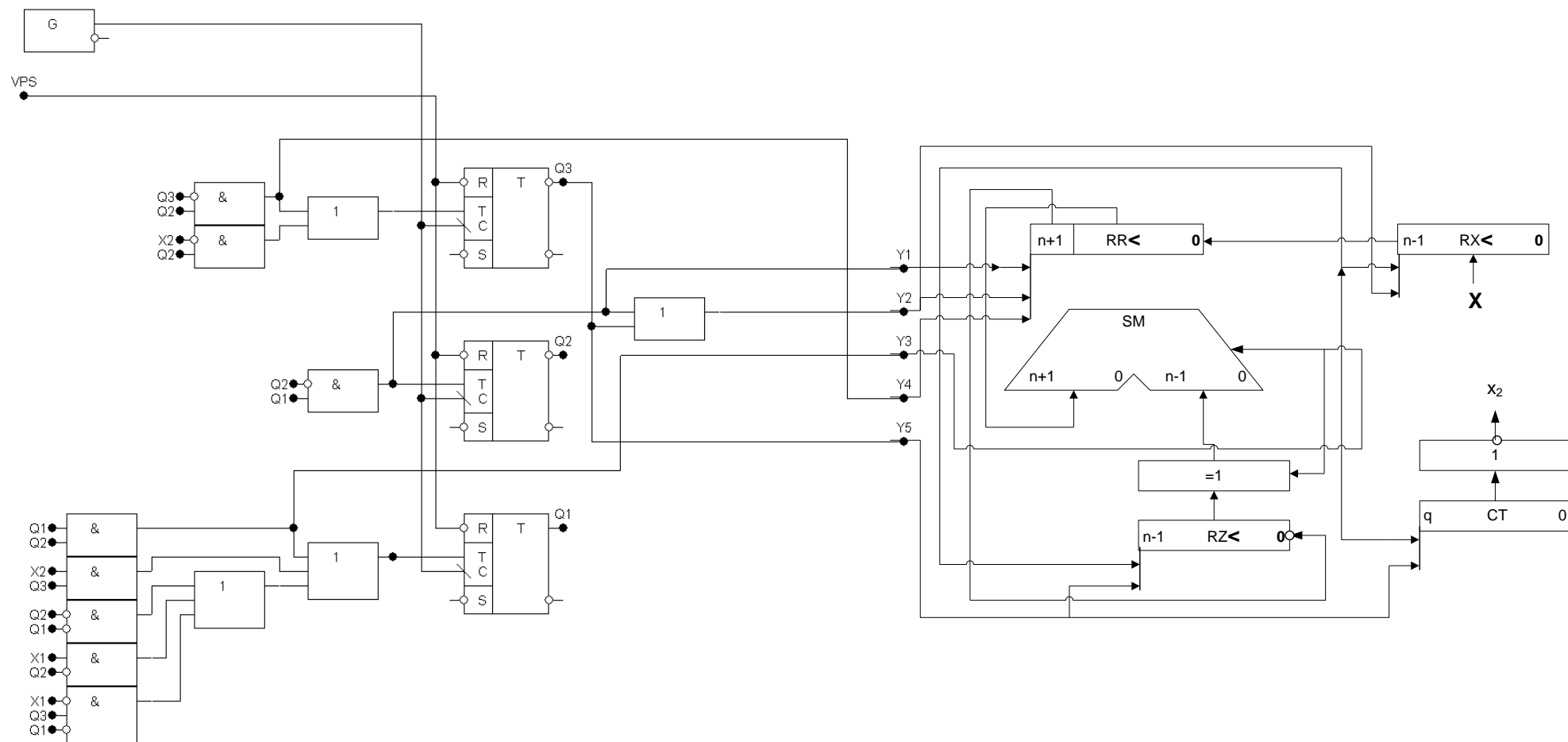


Рис . 5.4 Управляющий автомат з функціональною схемою