НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

# "КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

#### ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

### Кафедра обчислювальної техніки

## РОЗРАХУНКОВА РОБОТА

з дисципліни "Комп’ютерна арифметика"

Виконав: Філик Ярослав

Група ІО-12, Факультет ІОТ,

Залікова книжка № 1229

Номер технічного завдання 10011001101

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис керівника)

Київ – 2012 р.

**I.Завдання:**

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

2.1 теоретичне обґрунтування способу; (0,2)

2.1 операційну схему; (0,2)

2.2 змістовний мікроалгоритм; (0,2)

2.3 таблицю станів регістрів (лічильника), довжина яких забезпечує одержання 15 основних розрядів мантиси результату; (1,5)

2.4 функціональну схему з відображенням управляючих сигналів; (0.5)

2.5 закодований мікроалгоритм (мікрооперації замінюються управл. сигналами); (0.3)

2.6 граф управляючого автомата Мура з кодами вершин; (0,5)

2.7 обробку порядків (показати у довільній формі); (0,5)

2.8 форму запису нормалізованого результату з плаваючою комою в пам’ять. (0,1)

Вказані пункти для операцій додавання та віднімання виконати для етапу нормалізації результату з урахуванням можливого нулевого результату. Інші дії до етапу нормалізації результату можна проілюструвати у довільній формі.

3. Для операції з номером  побудувати управляючий автомат Мура на тригерах (тип вибрати самостійно) і елементах булевого базису. (1,5)

**II.Обгрунтування варіанту:**

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

 і ,

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

122910=100110011012

Х2= -10011101,0101101

Y2= +10101,1010101011

**III.Ocновна частина:**

**Завдання 1**

=1.10011101,0101101;

=0.10101,1010101011;

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

X2:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |

Y2:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

Представлення чисел у формі з характеристикою та мантисою:

Eх=8+27=13610=100010002

Еy=5+27=13310=100001012

X2:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |

Y2:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

**Завдання 2**

**2.1 Перший спосіб множення.**

**2.1.1 Теоретичне обґрунтування першого способу множення:**

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

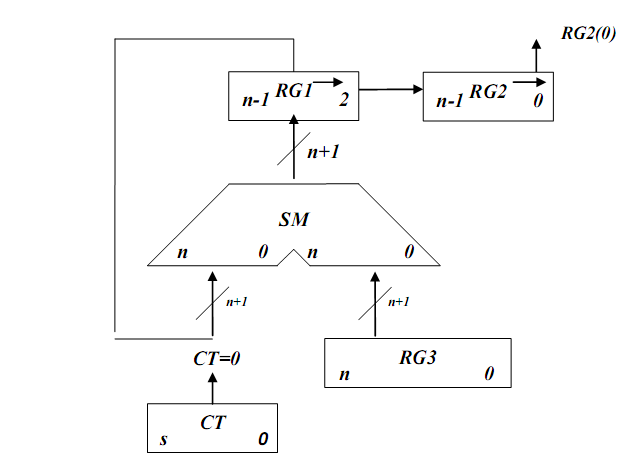
Множення мантис першим способом здійснюється з молодших розрядів множника, сума часткових добутків зсувається вправо,а множене залишається нерухомим. Тоді добуток двох чисел представляється у вигляді:

Z=YX*n*2*-n*+YX*n-1*2*-n+1*…+ YX12-1;

Z=(((0+YXn)2-1+ YXn-1)2-1…+ YX1)2-1;

Z=∑ni=1(Zi-1+YXn-i+1)2-1;

**2.1.2 Операційна схема:**



*Рисунок 2.1.1- Операційна схема.*

**2.1.3 Змістовний мікроалгоритм:**

1

1

Початок

RG1:=0; RG2:=X; RG3:=Y; CT:=15;

RG2[0]

RG1:=RG1+RG3;

RG1:=0.r(RG1); RG2:=RG1[0].r(RG2); CT:=CT-1;

CT=0

Кінець

0

0

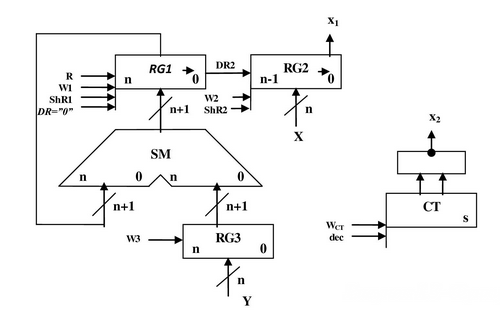
*Рисунок 2.1.2- Змістовний мікроалгоритм.*

**2.1.4 Таблиця станів регістрів:**

*Таблиця 2.1.1-Таблиця станів регістрів.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **RG1** | **RG2** | **RG3** | **CT** |
| **пс** | 0 | 100111010101101 | 101011010101011 | 1111 |
| **1** | +101011010101011  0101011010101011  0010101101010101 | 110011101010110 |  | 1110 |
| **2** | 0001010110101010 | 111001110101011 |  | 1101 |
| **3** | +101011010101011  0110110001010101  0011011000101010 | 111100111010101 |  | 1100 |
| **4** | +101011010101011  1000110011010101  0100011001101010 | 111110011101010 |  | 1011 |
| **5** | 0010001100110101 | 011111001110101 |  | 1010 |
| **6** | +101011010101011  0111100111100000  0011110011110000 | 001111100111010 |  | 1001 |
| **7** | 0001111001111000 | 000111110011101 |  | 1000 |
| **8** | +101011010101011  0111010100100011  0011101010010001 | 100011111001110 |  | 0111 |
| **9** | 0001110101001000 | 110001111100111 |  | 0110 |
| **10** | +101011010101011  0111001111110011  0011100111111001 | 111000111110011 |  | 0101 |
| **11** | +101011010101011  1001000010100100  0100100001010010 | 011100011111001 |  | 0100 |
| **12** | +101011010101011  1001111011111101  0100111101111110 | 101110001111100 |  | 0011 |
| **13** | 0010011110111111 | 010111000111110 |  | 0010 |
| **14** | 0001001111011111 | 101011100011111 |  | 0001 |
| **15** | +101011010101011  0110101010001010  **0011010101000101** | **010101110001111** |  | 0000 |

**2.1.5 Функціональна схема:**

****

*Рисунок 2.1.3 - Функціональна схема.*

**2.1.6 Закодований мікроалгоритм**

*Таблиця 2.1.2 - Таблиця кодування операцій і логічних умов.*

|  |  |  |  |
| --- | --- | --- | --- |
| Кодування мікрооперацій | | Кодування логічних умов | |
| МО | УС | ЛУ | Позначення |
| G1:=0  RG2:=X  RG3:=Y  CT:=15  RG1:=RG1+RG3  RG1:=0.r(RG1)  RG2:=RG1[0].r(RG2)  CT:=CT-1 | R  W2  W3  WCT  W1  ShR1  ShR2  dec | RG2[0]  CT=0 | X1  X2 |

Початок

R, W2, W3, WCT

X1

W1

ShR1,ShR2,dec

X2

Кінець

1

Z1

Z2

0

1

Z3

Z45

0

Z5

*Рисунок 2.1.4-Закодований мікроалгоритм.*

**2.1.7 Граф управляючого автомата Мура:**



**2.1.8 Обробка порядків:**

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків: Pz=Py+Py

Px=8; Py=5; Pz=1310=11012

**2.1.9 Нормалізація результату:**

Отримали результат: **0011010101000101010101110001111**

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

Порядок понижаємо на 2:

**11010101000101010101110001111**

Pz=11;

Запишемо нормалізованийрезультат:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

**2.2 Другий спосіб множення.**

**2.2.1 Теоретичне обґрунтування другого способу множення:**

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

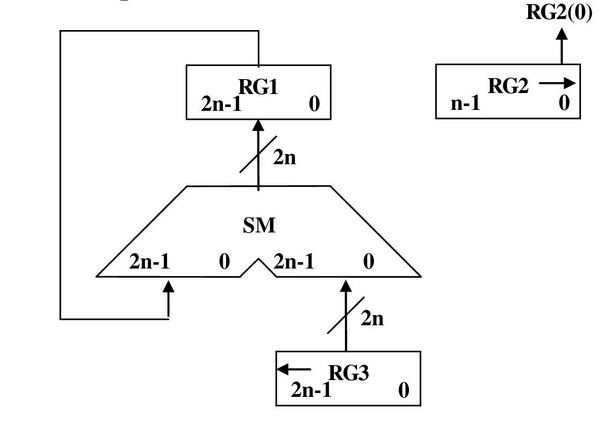
Множення мантис другим способом здійснюється з молодших розрядів, множене зсувається вліво, а сума часткових добутків залишається нерухомою.

Z=YXn2-n+YXn-12-n+1…+YX12-1;

Z=((0+YXn2-n)+YXn-1­2-n+1)…+YX12-1;

Z=∑ni=1Zi-1+YXn-i+12-1-n+i; Z0=0;Y0=0;

**2.2.2 Операційна схема:**



*Рисунок 2.2.1- Операційна схема.*

**2.2.3 Змістовний мікроалгоритм:**

01

1

RG1:=0; RG2:=X;RG3:=Y;

RG2[0]

RG1:=RG1+RG3;

RG2:=0.r(RG2); RG3:=l(RG1).0;

RX=0

Початок

Кінець

01

101

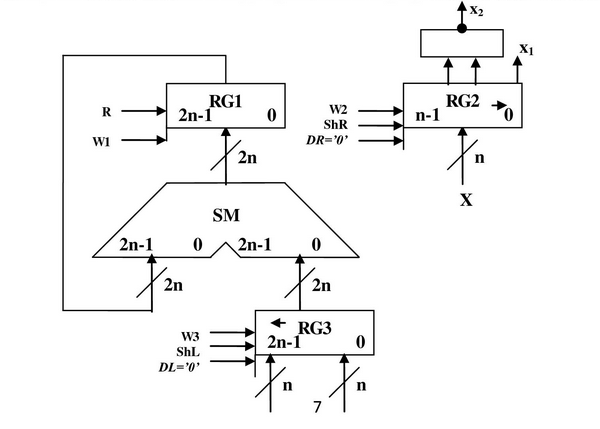
*Рисунок 2.2.2 - Змістовний мікроалгоритм.*

**2.2.4 Таблиця станів регістрів:**

*Таблиця2.2.1-Таблиця станіврегістрів.*

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **RG1** | **RG3** | **RG2** |
| **пс** | 0 | 000000000000000101011010101011 | 100111010101101 |
| **1** | +000000000000000101011010101011  000000000000000101011010101011 | 000000000000001010110101010110 | 010011101010110 |
| **2** | 000000000000000101011010101011 | 000000000000010101101010101100 | 001001110101011 |
| **3** | +000000000000010101101010101100  000000000000011011000101010111 | 000000000000101011010101011000 | 000100111010101 |
| **4** | +000000000000101011010101011000  000000000001000110011010101111 | 000000000001010110101010110000 | 000010011101010 |
| **5** | 000000000001000110011010101111 | 000000000010101101010101100000 | 000001001110101 |
| **6** | +000000000010101101010101100000  000000000011110011110000001111 | 000000000101011010101011000000 | 000000100111010 |
| **7** | 000000000011110011110000001111 | 000000001010110101010110000000 | 000000010011101 |
| **8** | +000000001010110101010110000000  000000001110101001000110001111 | 000000010101101010101100000000 | 000000001001110 |
| **9** | 000000001110101001000110001111 | 000000101011010101011000000000 | 000000000100111 |
| **10** | +000000101011010101011000000000  000000111001111110011110001111 | 000001010110101010110000000000 | 000000000010011 |
| **11** | +000001010110101010110000000000  000010010000101001001110001111 | 000010101101010101100000000000 | 000000000001001 |
| **12** | +000010101101010101100000000000  000100111101111110101110001111 | 000101011010101011000000000000 | 000000000000100 |
| **13** | 000100111101111110101110001111 | 001010110101010110000000000000 | 000000000000010 |
| **14** | 000100111101111110101110001111 | 010101101010101100000000000000 | 000000000000001 |
| **15** | +010101101010101100000000000000  **011010101000101010101110001111** | 101011010101011000000000000000 | 000000000000000 |

**2.2.5Функціональна 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  RG2:=X  RG3:=Y  RG1:=RG1+RG3  RG2:=0.r(PG2)  RG3:=l(RG3).0 | R  W2  W3  W1  ShR1  ShR2 | RG2[0]  RG2=0 | X1  X2 |

Z1

Початок

Z2

R,W2,W3

0

Х1

1

Z3

W1

ShR1, ShR2

Z4

0

Х2

1

Z6

Кінець

*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:Pz=Px+Py;

Px=8; Py=5; Pz=1310=11012

**2.2.9 Нормалізація результату:**

Отримали результат: **011010101000101010101110001111**

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

Порядок понижаємо на 1:

**11010101000101010101110001111**; =12;

Запишемо нормалізований результат:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

**2.3 Третій спосіб множення.**

**2.3.1 Теоретичне обгрунтування третього способу множення:**

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

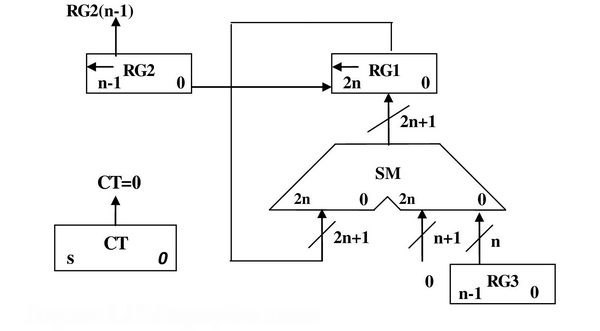
Множення мантис третім способом здійснюється зі старших розрядів множника, сума часткових добутків і множник зсуваються вліво, а множене нерухоме.

Z=YXn2-n+YXn-12-n+1…+YX12-1;

Z=YXn2-n+2(YXn-12-n+2(YXn-22-n…+2YX12-n));

Z=∑ni=1 2Zi-1+YXi2-n; Z0=0;Y0=0;

**2.3.2 Операційна схема:**



*Рисунок 2.3.1-Операційна схема*

**2.3.3 Змістовний мікроалгоритм:**

RG1:=0; RG2:=X; RG3:=Y; CT:=n;

RG2[n-1]

RG1:=RG1+RG3;

RG1:=l(RG1).0; RG2:=l (RG2).0; CT:=CT-1;

Початок

1

01

CT=0

Кінець

1

01

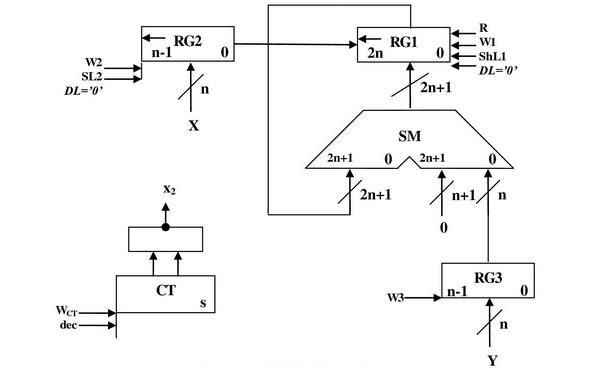
*Рисунок 2.3.2***-** *Змістовний мікроалгоритм.*

**2.3.4 Таблиця станів регістрів:**

*Таблиця 2.3.1- Таблиця станів регістрів*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **RG1** | **RG2** | **RG3** | **CT** |
| **пс** | 0000000000000000000000000000000 | 100111010101101 | 101011010101011 | 1111 |
| **1** | +0000000000000000100111010101101  0000000000000000100111010101101  0000000000000001001110101011010 |  | 001110101011010 | 1110 |
| **2** | 0000000000000010011101110110100 |  | 011101010110100 | 1101 |
| **3** | 000000000000101011010101011000 |  | 111010101101000 | 1100 |
| **4** | +000000000000000101011010101011  000000000000110000110000000011  000000000001100001100000000110 |  | 110101011010000 | 1011 |
| **5** | +000000000000000101011010101011  000000000001100110111010110001  000000000011001101110101100010 |  | 101010110100000 | 1010 |
| **6** | +000000000000000101011010101011  000000000011010011010000001101  000000000110100110100000011010 |  | 010101101000000 | 1001 |
| **7** | 000000001101001101000000110100 |  | 101011010000000 | 1000 |
| **8** | +000000000000000101011010101011  000000001101010010011011011111  000000011010100100110110111110 |  | 010110100000000 | 0111 |
| **9** | 000000110101001001101101111100 |  | 101101000000000 | 0110 |
| **10** | +000000000000000101011010101011  000000110101001111001000100111  000001101010011110010001001110 |  | 011010000000000 | 0101 |
| **11** | 000011010100111100100010011100 |  | 110100000000000 | 0100 |
| **12** | +000000000000000101011010101011  000011010101000001111101000111  000110101010000011111010001110 |  | 101000000000000 | 0011 |
| **13** | +000000000000000101011010101011  000110101010001001010100111001  001101010100010010101001110010 |  | 010000000000000 | 0010 |
| **14** | 011010101000100101010011100100 |  | 100000000000000 | 0001 |
| **15** | +000000000000000101011010101011  011010101000101010101110001111 |  | 000000000000000 | 0 |

**2.3.5 Функціональна схема:**



*Рисунок 2.3.3* **-** *Функціональна схема.*

**2.3.6 Закодований мікроалгоритм:**

*Таблиця 2.3.2-Таблиця кодування операцій і логічних умов.*

|  |  |  |  |
| --- | --- | --- | --- |
| Кодування мікрооперацій | | Кодування логічних умов | |
| МО | УС | ЛУ | Позначення |
| RG1:=0  RG2:=X  RG3:=Y  CT:=15  RG1:=RG1+RG3  RG1:=l(RG1).0  RG2:=l(RG2).0  CT:=CT-1 | R  W2  W3  WCT  W1  ShL1  ShL2  dec | RG2[n-1]  CT=0 | X1  X2 |

Початок

R, W2, W3, WCT

X1

W1

ShL1,ShL2,dec

X2

Кінець

1

0

1

0

Z2

Z3

Z4

Z5

Z1

*Рисунок 2.3.4-Закодований мікроалгоритм.*

**2.3.7 Граф управляючого автомата Мура з кодами вершин:**



*Рисунок 2.3.5-Граф автомата Мура*

**2.3.8 Обробка порядків:**

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків: Pz=Py+Py

Px=8; Py=5; Pz=1310=11012

**2.3.9 Нормалізація результату:**

Отримали результат: 011010101000101010101110001111

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

Порядок понижаємо на 1:

11010101000101010101110001111; Pz=12;

Запишемо нормалізований результат:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

**2.4 Четвертий спосіб множення.**

**2.4.1 Теоретичне обґрунтування четвертого способу множення:**

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

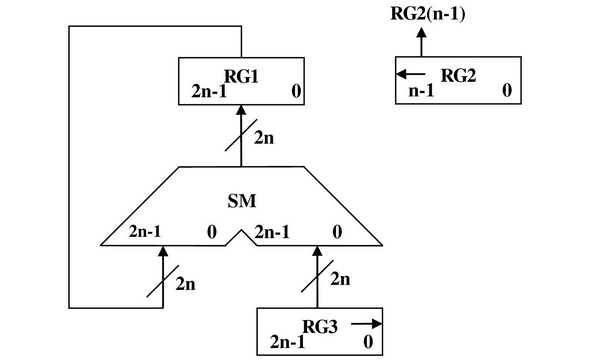
Множення здійснюється зі старших розрядів множника, сума часткових добутків залишається нерухомою,множене зсувається праворуч, множник ліворуч.

.

*.*

з початковими значеннями i=1,Y0=2-1Y,Z0=0.

**2.4.2 Операційна схема:**



*Рисунок 2.4.1-Операційна схема*

**2.4.3 Змістовний мікроалгоритм:**

Початок

RG1:=0; RG2:=X; RG3:=Y; RG3:=0.r(RG3)

01

RG2[n-1]

1

RG1:=RG1+RG3;

RG3:=0.r(RG3)

RG2:=l(RG2).0

01

RG2=0

1

Кінець

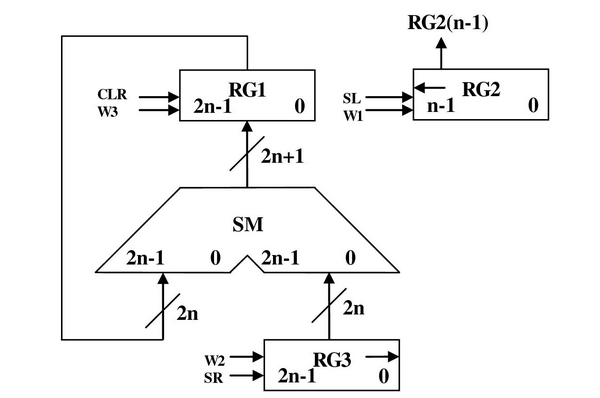
*Рисунок 2.4.2***-** *Змістовний мікроалгоритм.*

**2.4.4 Таблиця станів регістрів:**

*Таблиця 2.4.1- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **RG1** | **RG3** | **RG2** |
| **пс** | 000000000000000000000000000000 | 010101101010101100000000000000 | 100111010101101 |
| **1** | +010101101010101100000000000000  010101101010101100000000000000 | 001010110101010110000000000000 | 001110101011010 |
| **2** | 010101101010101100000000000000 | 000101011010101011000000000000 | 011101010110100 |
| **3** | 010101101010101100000000000000 | 000010101101010101100000000000 | 111010101101000 |
| **4** | +000010101101010101100000000000  011000011000000001100000000000 | 000001010110110010110000000000 | 110101011010000 |
| **5** | +000001010110101010110000000000  011001101110101100010000000000 | 000000101011010101011000000000 | 101010110100000 |
| **6** | +000000101011010101011000000000  011010011010000001101000000000 | 000000010101101010101100000000 | 010101101000000 |
| **7** | 011010011010000001101000000000 | 000000001010110101010110000000 | 101011010000000 |
| **8** | +000000001010110101010110000000  011010100100110110111110000000 | 000000000101011010101011000000 | 010110100000000 |
| **9** | 011010100100110110111110000000 | 000000000010101101010101100000 | 101101000000000 |
| **10** | +000000000010101101010101100000  011010100111100100010011100000 | 000000000001010110101010110000 | 011010000000000 |
| **11** | 011010100111100100010011100000 | 000000000000101011010101011000 | 110100000000000 |
| **12** | +000000000000101011010101011000  011010101000001111101000111000 | 000000000000010101101010101100 | 101000000000000 |
| **13** | +000000000000010101101010101100  011010101000100101010011100100 | 000000000000001010110101010110 | 010000000000000 |
| **14** | 011010101000100101010011100100 | 000000000000000101011010101011 | 100000000000000 |
| **15** | +000000000000000101011010101011  **011010101000101010101110001111** | 000000000000000010101101010101 | 000000000000000 |

**2.4.5Функціональна схема:**



*Рисунок2.4.3* **-** *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:=l(RG2).0 | R  W2  W3  W1  ShR  ShL | RG2[n-1]  RG2=0 | X1  X2 |

R, W2, W3, ShR

X1

W1

ShR,ShL

X2

Кінець

1

0

1

0

Z2

Z3

Z4

Z5

Z1

Початок

*Рисунок 2.4.4-Закодований мікроалгоритм.*

**2.4.7Граф 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:

=8; =5; =1310=11012

**2.4.9 Нормалізація результату:**

Отримали результат: **011010101000101010101110001111**

Знак мантиси: 1 0 = 1.

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

Порядок понижаємо на 1:

**11010101000101010101110001111**;=12;

Запишемо нормалізований результат:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

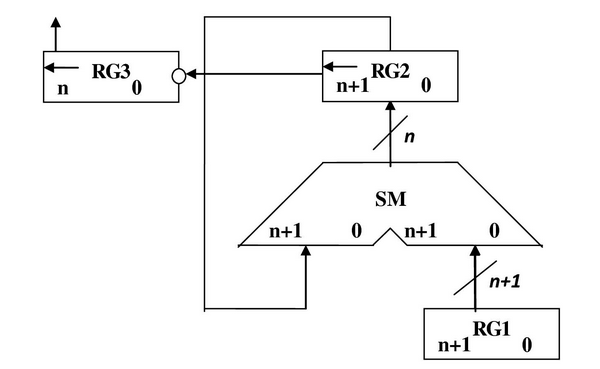
**2.5. Першиий спосіб ділення.**

**2.5.1Теоритичне обґрунтування першого способу ділення:**

Нехай ділене Х і дільник Y є n-розрядними правильними дробами, поданими в прямому коді. В цьому випадку знаковий та основні розряди операндів обробляються окремо. Знак результату визначається шляхом підсумовування по модулю 2 цифр, записаних в знакових розрядах.

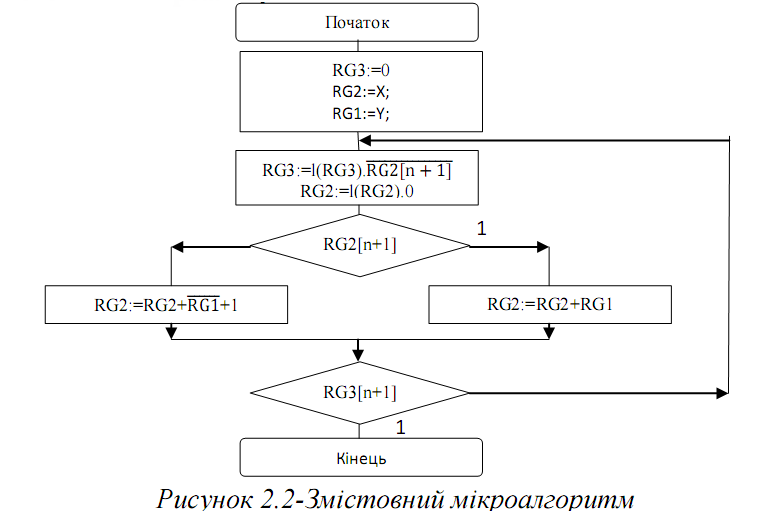
При реалізації ділення за першим методом здійснюється зсув вліво залишку при нерухомому дільнику. Черговий залишок формується врегістріР2(у вихідному станів в цьому регістрі записаний Х). Виходи Р2 підключені до входів СМ безпосередньо, тобто ланцюги видачі коду з Р2 не потрібні. Час для підключення n+1 цифри частки визначається виразом t=(n+1)(tt+tc), де tt- тривалість виконання мікрооперації додавання-віднімання; tc - тривалість виконання мікрооперації зсуву.

**2.5.2 Операційна схема:**



*Рисунок 2.5.1-Операційна схема*

**2.5.3 Змістовний мікроалгоритм:**

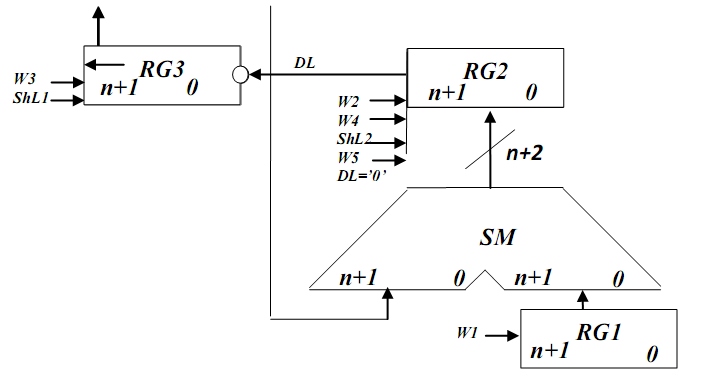


*Рисунок2.5.2-Змістовний мікроалгоритм*

**2.5.4 Таблиця станів регістрів:**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **RG3(Z)** | **RG2(X)** | **RG1(Y)** |
| **пс** | 000000000000000 | 00100111010101101 | 00101011010101011пк  11010100101010101дк |
| **1** | 0000000000000001 | 01001110101011010  +11010100101010101  =00100011010101111 |  |
| **2** | 0000000000000011 | 01000110101011110  +11010100101010101  =00011011010110011 |  |
| **3** | 0000000000000111 | 00110110101100110  +11010100101010101  =00001011010111011 |  |
| **4** | 0000000000001111 | 00010110101110110  +11010100101010101  =11101011011001011 |  |
| **5** | 0000000000011110 | 11010110110010110  +00101011010101011  =00000010001000001 |  |
| **6** | 0000000000111101 | 00000100010000010  +11010100101010101  =11011000111010111 |  |
| **7** | 0000000001111010 | 10110001110101110  +00101011010101011  =11011101001011001 |  |
| **8** | 0000000011110100 | 10111010010110010  +00101011010101011  =11100101101011101 |  |
| **9** | 0000000111101000 | 11001011010111010  +00101011010101011  =11110110101100101 |  |
| **10** | 0000001111010000 | 11101101011001010  +00101011010101011  =00011000101110101 |  |
| **11** | 0000011110100001 | 00110001011101010  +11010100101010101  =00000110000111111 |  |
| **12** | 0000111101000011 | 00001100001111110  +11010100101010101  =11100000111010011 |  |
| **13** | 0001111010000111 | 11000001110100110  +00101011010101011  =11101101001010001 |  |
| **14** | 0011110100001110 | 11011010010100010  +00101011010101011  =00000101101001101 |  |
| **15** | 0111101000011101 | 00001011010011010  +11010100101010101  =11011111111101111 |  |
| **16** | **1111010000111010** | 10111111111011110  +00101011010101011  =11101011010001001 |  |

**2.5.5Функціональна схема:**

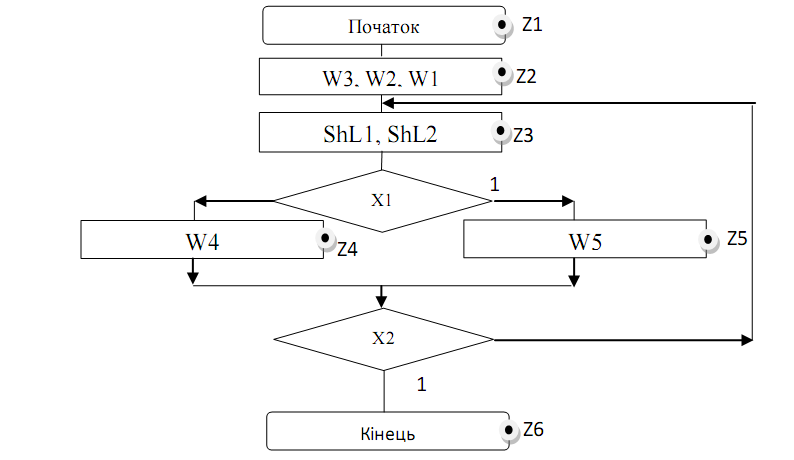


*Рисунок 2.5.3 – Функціональна схема*

**2.5.6 Закодований мікроалгоритм**

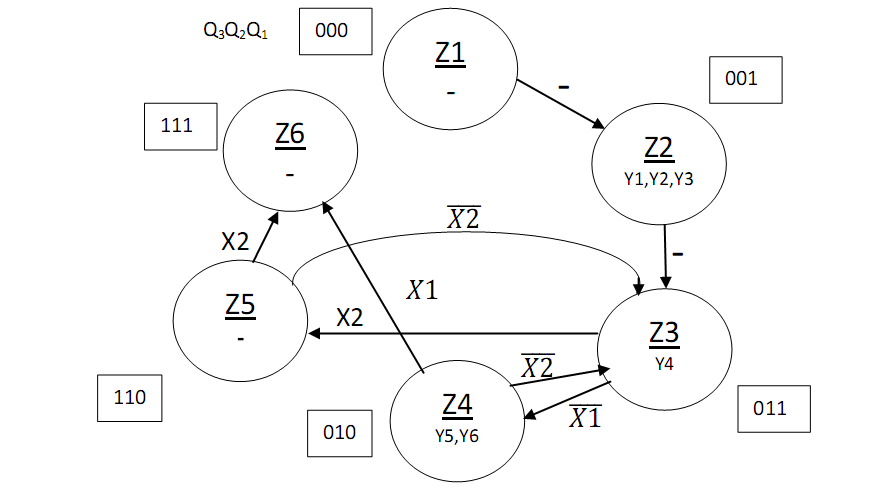
*Таблиця 2.5.2-Таблиця кодування операцій і логічних умов.*

|  |  |  |  |
| --- | --- | --- | --- |
| Кодування мікрооперацій | | Кодування логічних умов | |
| МО | УС | ЛУ | Позначення |
| RG3:=0  RG2:=X;  RG1:=Y;  RG3:=l(RG3).RG2[n+1]  RG2:=l(RG2).0  RG2:=RG2+RG1+1  RG2:=RG2+RG1 | W3  W2  W1  ShL1  ShL2  W4  W5 | RG2[n-1]  RG2=0 | X1  X2 |



*Рисунок 2.5.4-Закодований мікроалгоритм.*

**2.5.7Граф управляючого автомата Мура з кодами вершин:**



*Рисунок 2.5.5 -Граф управляючого автомата.*

**2.5.8 Обробка порядків:**

Порядок частки буде дорівнювати:

В моєму випадку =8; =5; =3;

**2.5.8 Нормалізація результату:**

Отримали результат: **1111010000111010**

Знак мантиси: 1 0 = 1.

Нормалізація мантиси не потрібна.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |

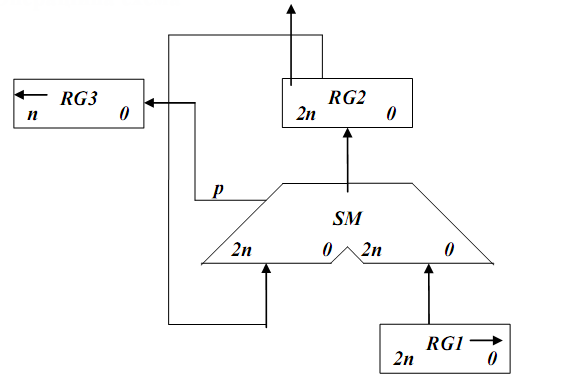
**2.6. Другий спосіб ділення.**

**2.6.1 Теоретичне обгрунтування другого способу ділення:**

Нехай ділене Х і дільник Y є n-розрядними правильними дробами, поданими в прямому коді. В цьому випадку знаковий та основні розряди операндів обробляються окремо. Знак результату визначається шляхом підсумовування по модулю 2 цифр, записаних в знакових розрядах.

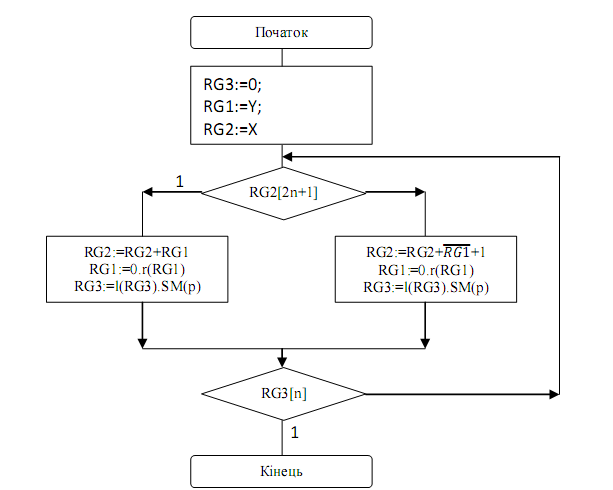
Остача нерухома, дільник зсувається праворуч. Як і при множенні з нерухомою сумою часткових добутків можна водночас виконувати підсумування і віднімання, зсув в регістрах Y,Z. Тобто 1 цикл може складатися з 1 такту, це дає прискорення відносно 1-го способу.

**2.6.2Операційнасхема**



*Рисунок2.6.1-Операційнасхема*

**2.6.3Змістовний мікроалгоритм**



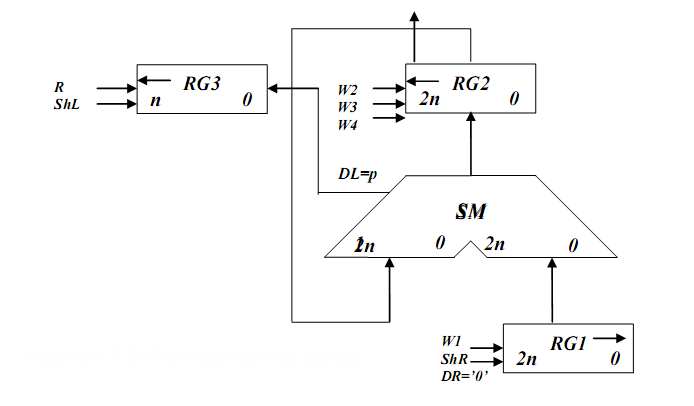
*Рисунок 2.6.2-Змістовний мікроалгоритм*

**2.6.4Таблицястаніврегістрів**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **RG3(Z)** | **RG2(X)** | **RG1(Y)** |
| **пс** | 0000000000000001 | 010011101010110100000000000000 | 001010110101010110000000000000 |
| **1** | 0000000000000011 | +110101001010101010000000000000  =001000110101011110000000000000 | 000101011010101011000000000000 |
| **2** | 0000000000000111 | +111010100101010101000000000000  =000011011010110011000000000000 | 000010101101010101100000000000 |
| **3** | 0000000000001111 | +111101010010101010100000000000  =000000101101011101100000000000 | 000001010110101010110000000000 |
| **4** | 0000000000011110 | +111110101001010101010000000000  =111111010110110010110000000000 | 000000101011010101011000000000 |
| **5** | 0000000000111101 | +000000101011010101011000000000  =000000000010001000001000000000 | 000000010101101010101100000000 |
| **6** | 0000000001111010 | +111111101010010101010100000000  =111111101100011101011100000000 | 000000001010110101010110000000 |
| **7** | 0000000011110100 | +000000001010110101010110000000  =111111110111010010110010000000 | 000000000101011010101011000000 |
| **8** | 0000000111101000 | +000000000101011010101011000000  =111111111100101101011101000000 | 000000000010101101010101100000 |
| **9** | 0000001111010000 | +000000000010101101010101100000  =111111111111011010110010100000 | 000000000001010110101010110000 |
| **10** | 0000011110100001 | +000000000001010110101010110000  =000000000000110001011101010000 | 000000000000101011010101011000 |
| **11** | 0000111101000011 | +111111111111010100101010101000  =000000000000000110000111111000 | 000000000000010101101010101100 |
| **12** | 0001111010000110 | +111111111111101010010101010100  =111111111111110000011101001100 | 000000000000001010110101010110 |
| **13** | 0011110100001100 | +000000000000001010110101010110  =111111111111111011010010100010 | 000000000000000101011010101011 |
| **14** | 0111101000011001 | +000000000000000101011010101011  =000000000000000000101101001101 | 000000000000000010101101010101 |
| **15** | **1111010000110010** | +111111111111111101010010101011  =111111111111111101111111111000 | 000000000000000001010110101010 |

*Таблиця 2.6.1- Таблиця станіврегістрів*

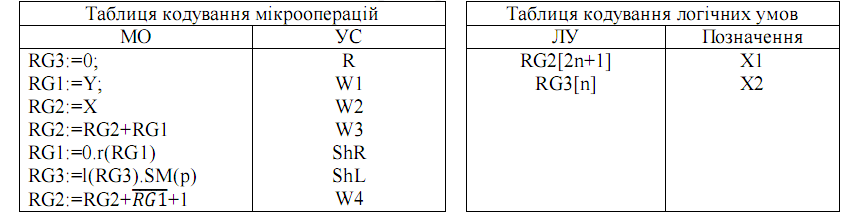
**2.6.5Функціональна схема з відображенням управляючих сигналів**

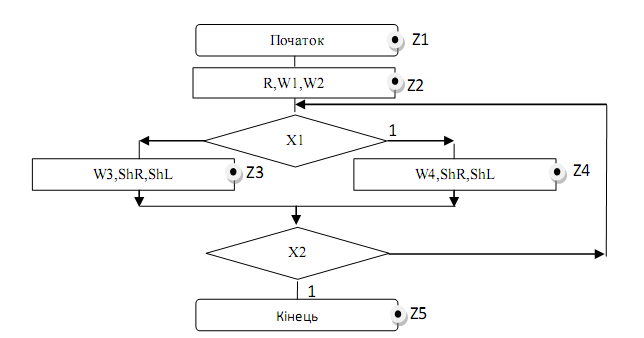


*Рисунок2.6.3-Функціональна схема*

**2.6.6Закодований мікроалгоритм**

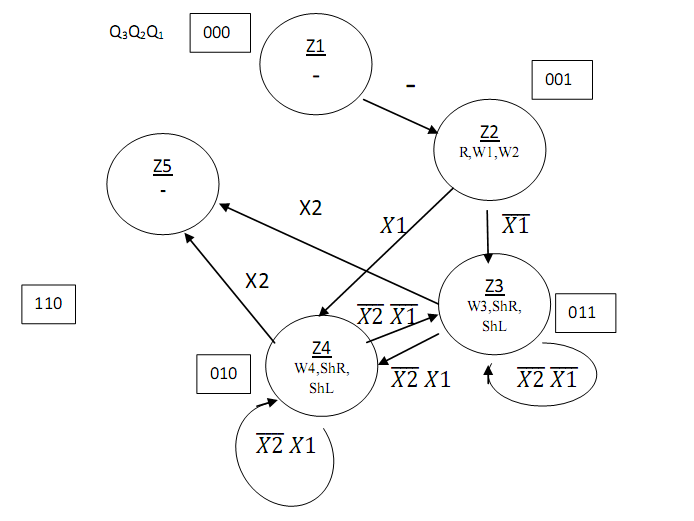
*Таблиця2.6.2- Таблиця кодування мікрооперацій*





*Рисунок2.6.4- Закодований мікроалгоритм*

**2.6.7 Граф управляючого автомата Мура з кодами вершин**



*Рисунок2.6.5- Граф автомата Мура*

**2.6.8 Обробка порядків:**

Порядок частки буде дорівнювати:

В моєму випадку =8; =5; =3;

**2.6.9 Нормалізація результату:**

Отримали результат: **1111010000110010**

Знак мантиси: 1 0 = 1.

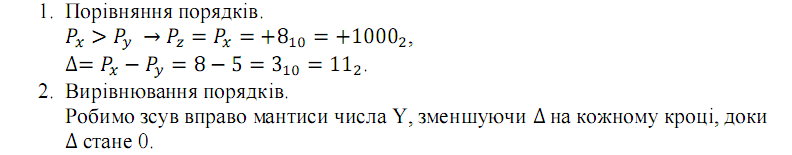
Нормалізація мантиси не потрібна.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 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 доповню вальних кодах, при необхідності числа у ДК переводяться в АЛП. Додавання виконується порозрядно на n-розрядному суматорі з переносом. Останній етап – нормалізація результату. Виконується за допомогою зсуву мантиси результату і коригування порядку результату. Порушення нормалізації можливо вліво і вправо,на 1розряд вліво і на n розрядів вправо.



*Таблиця 2.7.1- Таблиця зсуву мантиси на етапі вирівнювання порядків*

|  |  |  |
| --- | --- | --- |
| MY | ∆ | Мікрооперація |
| 0.101011010101011 | 11 | П.С. |
| 0.010101101010101 | 10 | ∆:=∆-1 |
| 0.001010110101010 | 01 | ∆:=∆-1 |
| 0.000101011010101 | 00 | ∆:=∆-1 |

3. Додавання мантис в ДК

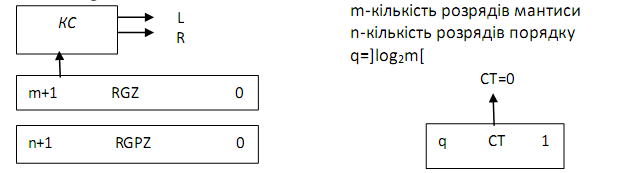
*Таблиця 2.7.2-Додавання мантис*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MX | 1 | 1, | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| MY | 0 | 0, | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| MZ | 1 | 1, | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

4. Нормалізація результату

ПК: Мz=1.101100110000010

**2.7.2Операційнасхема**



*Рисунок2.7.1-Операційна схема*

Виконаємо синтез КС для визначення порушення нормалізації.

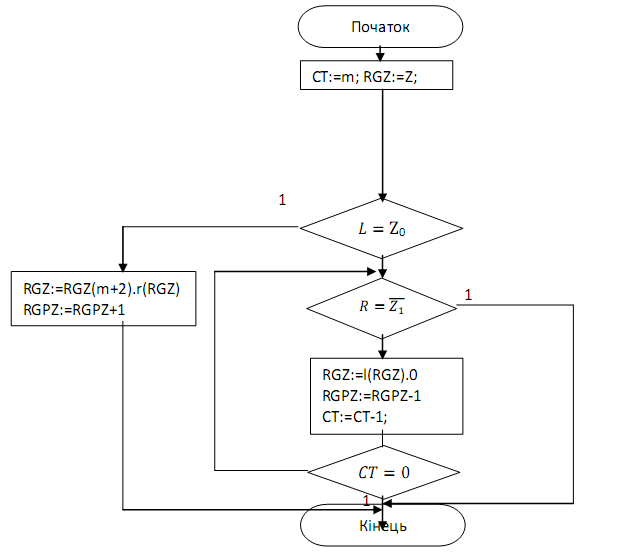
*Таблиця 2.7.3-Визначенняпорушеннянормалізації*

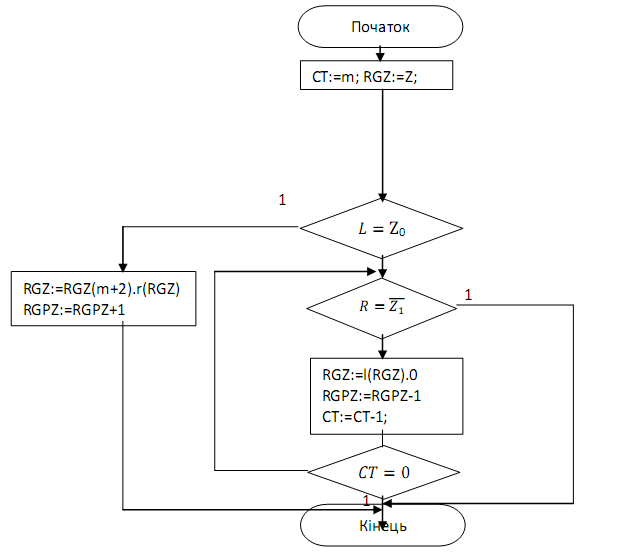
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Розрядирегістру  RGZ | | | Значення  функцій | |
| Z’0 | Z0 | Z1 | L | R |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |



Результат беремо по модулю, знак встановлюємо за Z’0 до нормалізації.

**2.7.5 Змістовний алгоритм**





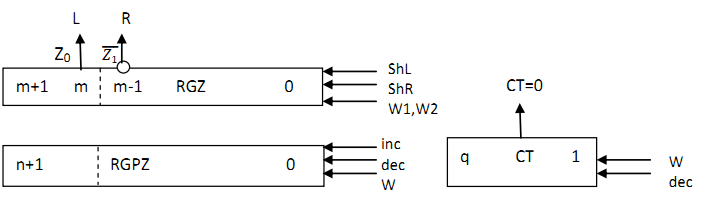
*Рисунок2.7.2-Змістовний мікроалгоритм*

**2.7.4Таблиця станів регістрів**

*Таблиця2.7.4- Таблиця станіврегістрів*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№**  **такту** | **RGPZ** | **RGZ** | **ЛПН(L)** | **ППН(R)** | **СT** | **Мікрооперація** |
| **ПС** | 001000 | 11.101100110000010 | 0 | 0 | 100 |  |

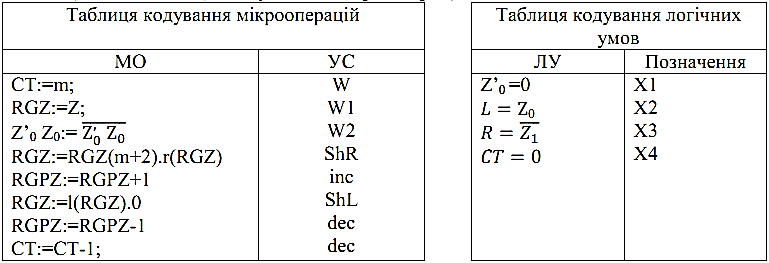
**2.7.5 Функціональна схема з відображенням керуючих сигналів**

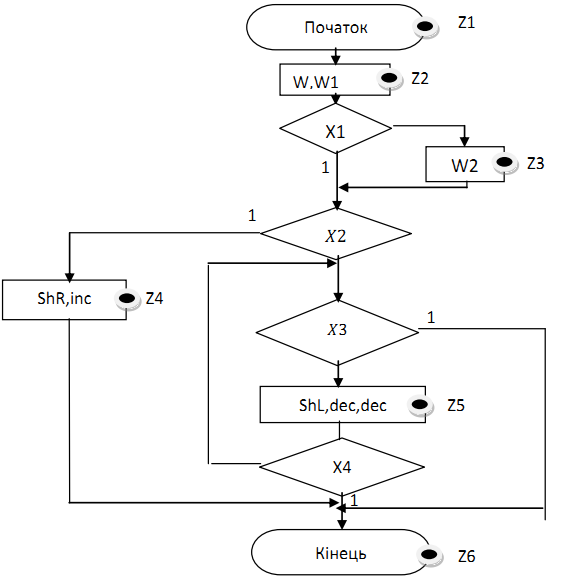
****

*Рисунок 2.7.3 – Функціональна схема*

**2.7.6 Закодований мікроалгоритм**

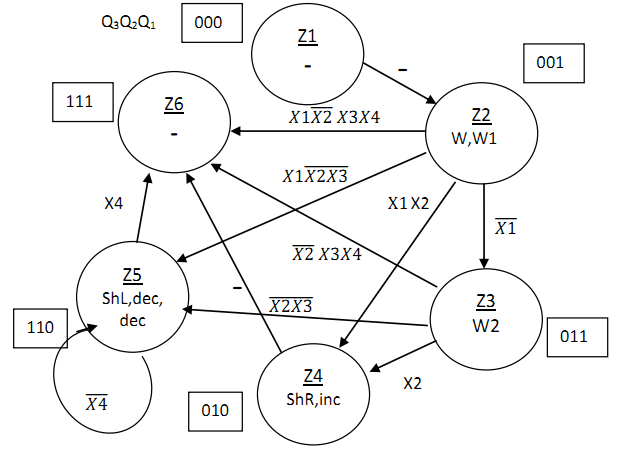
*Таблиця 2.7.5 – Таблиця кодування*

****

****

*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**

PZ=10002

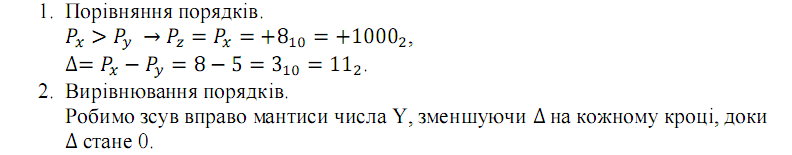
**2.7.9 Форма запису результату з плаваючою комою**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

**2.8. Операція віднімання чисел.**

**2.8.1 Теоретичне обґрунтування способу**

Операція віднімання виконується за тим же алгоритмом що і операція додавання. Різниця в тому, що на третьому кроці замість операції додавання мантисс виконується операція віднімання.



*Таблиця 2.7.1- Таблиця зсуву мантиси на етапі вирівнювання порядків*

|  |  |  |
| --- | --- | --- |
| MY | ∆ | Мікрооперація |
| 0.101011010101011 | 11 | П.С. |
| 0.010101101010101 | 10 | ∆:=∆-1 |
| 0.001010110101010 | 01 | ∆:=∆-1 |
| 0.000101011010101 | 00 | ∆:=∆-1 |

3. Віднімання мантис у модифікованому ДК.

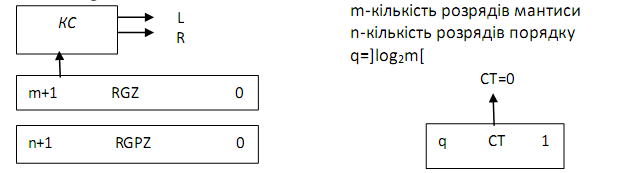
*Таблиця 2.7.2-Віднімання мантис*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MX | 1 | 1, | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| MY | 1 | 1, | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| MZ | 1 | 1, | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |

4. Нормалізація результату

ПК: Мz=1.100001111011001

**2.7.2Операційнасхема**



*Рисунок2.7.1-Операційна схема*

Виконаємо синтез КС для визначення порушення нормалізації.

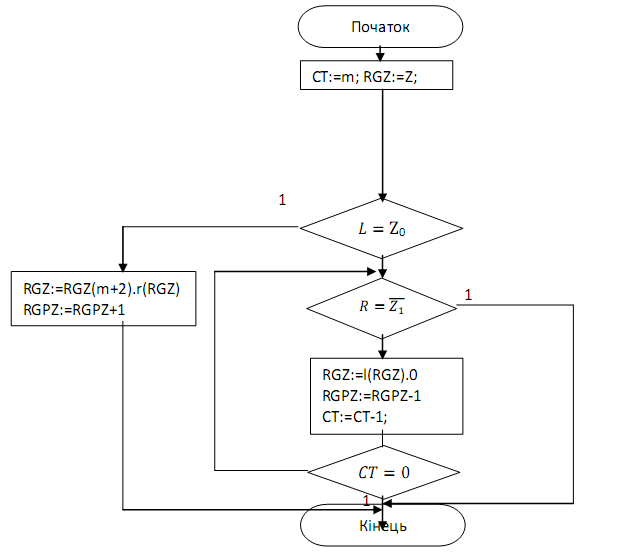
*Таблиця 2.7.3-Визначенняпорушеннянормалізації*

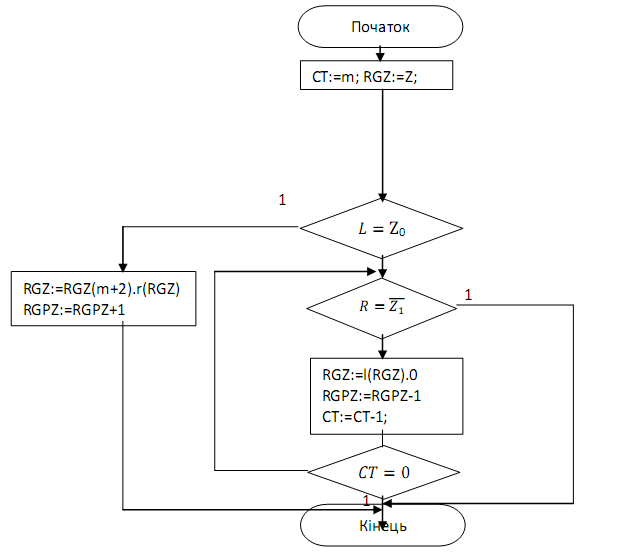
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Розрядирегістру  RGZ | | | Значення  функцій | |
| Z’0 | Z0 | Z1 | L | R |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |



Результат беремо по модулю, знак встановлюємо за Z’0 до нормалізації.

**2.7.5 Змістовний алгоритм**





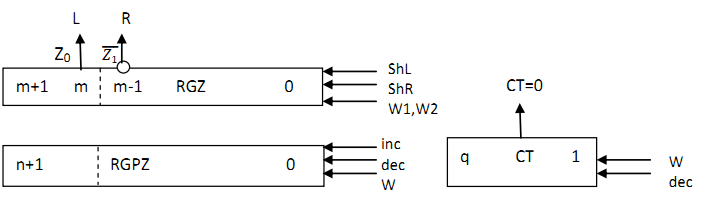
*Рисунок2.7.2-Змістовний мікроалгоритм*

**2.7.4 Таблиця станів регістрів**

*Таблиця2.7.4- Таблиця станів регістрів*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№**  **такту** | **RGPZ** | **RGZ** | **ЛПН(L)** | **ППН(R)** | **СT** | **Мікрооперація** |
| **ПС** | 001000 | 11.100001111011001 | 0 | 0 | 100 |  |

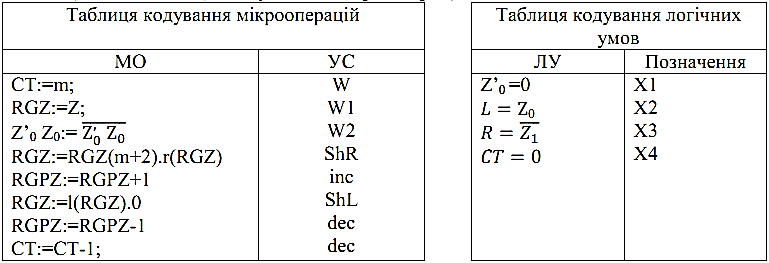
**2.7.5 Функціональна схема з відображенням керуючих сигналів**

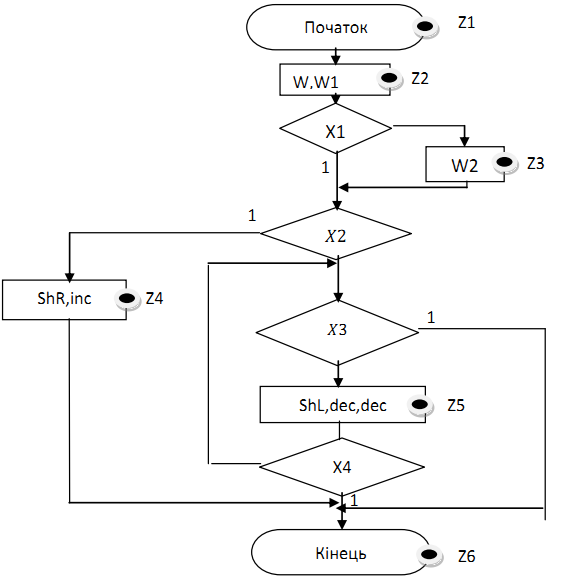
****

*Рисунок 2.7.3 – Функціональна схема*

**2.7.6 Закодований мікроалгоритм**

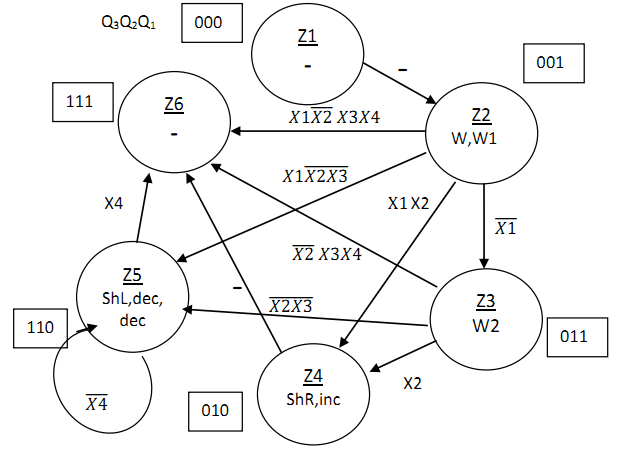
*Таблиця 2.7.5 – Таблиця кодування*

****

****

*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**

PZ=10002

**2.7.9 Форма запису результату з плаваючою комою**1.100001111011001

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |

**Завдання 3**

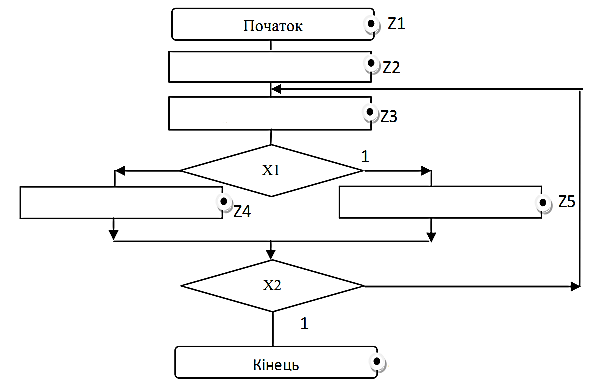
**Синтез управляючого автомату Мура на тригерах для операції ділення першим способом**

**3.1 Таблиця кодування сигналів**

*Таблиця 3.1 – Таблиця кодування сигналів*

|  |  |
| --- | --- |
| W3, W2, W1 | Y1 |
| ShL1, ShL2 | Y2 |
| W4 | Y3 |
| W5 | Y4 |

**3.2 Мікроалгоритм в термінах управляючого автомата**

****

**Z11**

Y4

Y3

Y2

Y1

*Рисунок 3.1 – Закодований мікроалгоритм*

**3.3 Граф автомата**



*Рисунок 3.2 – Граф циклічного автомата*

**3.4 Таблиця переходів циклічного автомата на D-тригерах**

*Таблиця 3.2 – Таблиця переходів*

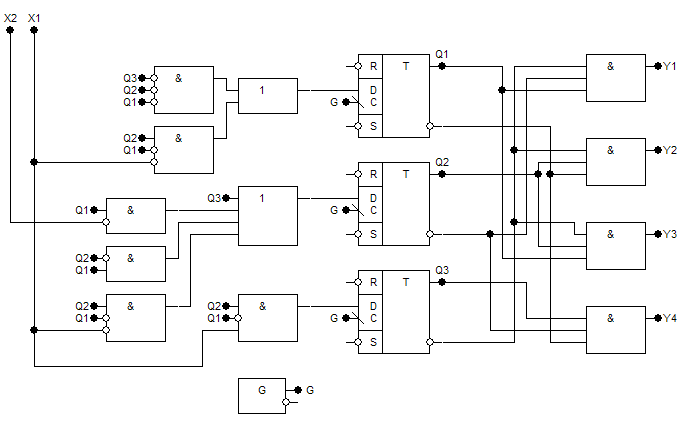
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Пер. | | Ст. ст. | Нов. стан | Вх. сигн. | Вих. сигн. | Функціїтригерів | | | |
| Q3Q2Q1 | Q3Q2Q1 | X1X2 | Y1Y2Y3Y4 | D3 | D2 | | D1 |
| Z1→Z2 | 000 | | 001 | -- | 0000 | 0 | | 0 | 1 |
| Z2→Z3 | 001 | | 010 | -- | 1000 | 0 | | 1 | 0 |
| Z3→Z4 | 010 | | 011 | 0- | 0100 | 0 | | 1 | 1 |
| Z3→Z5 | 010 | | 100 | 1- | 0100 | 1 | | 0 | 0 |
| Z4→Z1 | 011 | | 000 | -1 | 0010 | 0 | | 0 | 0 |
| Z4→Z3 | 011 | | 010 | -0 | 0010 | 0 | | 1 | 0 |
| Z5→Z1 | 100 | | 000 | -1 | 0001 | 0 | | 0 | 0 |
| Z5→Z3 | 100 | | 010 | -0 | 0001 | 0 | | 1 | 0 |

**3.5 Мінімізація функцій тригерів**



*Рисунок 3.3 –Мінімізаціяфункційтригерів*

**3.6 Функціональна схема автомата**



D

D

D

*Рисунок 3.4 - Функціональна схема*