НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

# "КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

#### ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

### Кафедра обчислювальної техніки

## РОЗРАХУНКОВА РАБОТА

з дисципліни "Комп’ютерна арифметика"

Виконав: Кукса Вячеслав Вікторович

Група ІО-12,

Факультет ІОТ,

Залікова книжка № 1214

Допущена до захисту \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Номер технічного завдання 10010111110

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

(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.

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;

2.1 операційну схему;

2.2 змістовний мікроалгоритм;

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

2.4 функціональну схему з відображенням управляючих сигналів;

2.5 закодований мікроалгоритм (мікрооперації замінюються управляючими сигналами);

2.6 граф управляючого автомата Мура з кодами вершин;

2.7 обробку порядків (показати у довільній формі);

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

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.

**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).

 =-10011011,1101110

 =+10101,0111101101

**III.Ocновна частина:**

**Завдання 1**

=1.10011011,1101110;

=0.10101,0111101101;

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

Px=+10002; Mx=-1001101111011102;

Py=+01012; My=+101010111101101;

X2:

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

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

Y2:

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

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

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

Ex = Px+2m = 810+27 = 810+12810 = 13610  = 100010002; Mx=-1001101111011102;

Ey = Py+2m = 510+27 = 510 + 12810 = 13310 = 100001012; My=+101010111101101;

X2:

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

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

Y2:

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

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

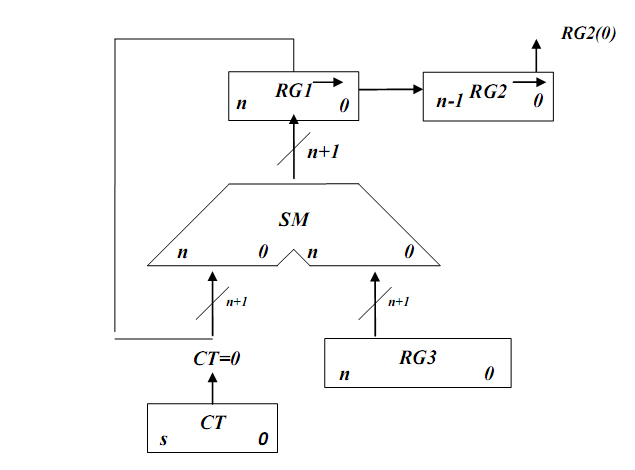
Множення мантис першим способом здійснюється з молодших розрядів множника, сума часткових добутків зсувається вправо, а множене залишається нерухомим. Тоді добуток двох чисел представляється у вигляді:

Z=Y+ Y…+ Y;

Z=(((0+Y)+ Y)…+ Y);

Z=;

**2.1.2 Операційна схема:**



*Рисунок 2.1.1- Операційна схема.*

**2.1.3 Змістовний мікроалгоритм:**

Початок

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

0

RG2[0]

1

RG1:=RG1+RG3

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

0

CT=0

Кінець

1

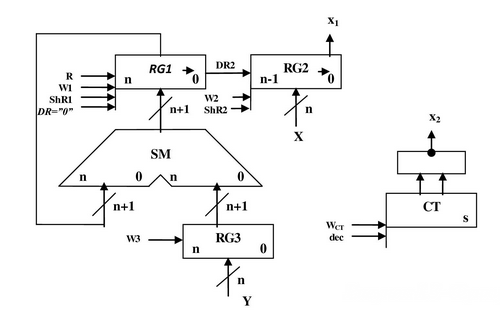
*Рисунок 2.1.2 - Змістовний мікроалгоритм.*

**2.1.4 Таблиця станів регістрів:**

*Таблиця 2.1.1-Таблиця станів регістрів.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | RG1 | RG2 | RG3 | CT |
| ПС | 0 | 100110111101110 | 0101010111101101 | 1111 |
| 1 | 0 | 010011011110111 | 0101010111101101 | 1110 |
| 2 | 0010101011110110 | 101001101111011 | 0101010111101101 | 1101 |
| 3 | +  0101010111101101  =  1000000011100011  0100000001110001 | 110100110111101 | 0101010111101101 | 1100 |
| 4 | +  0101010111101101  =  1001011001011110  0100101100101111 | 011010011011110 | 0101010111101101 | 1011 |
| 5 | 0010010110010111 | 101101001101111 | 0101010111101101 | 1010 |
| 6 | +  0101010111101101  =  0111101110000100  0011110111000010 | 010110100110111 | 0101010111101101 | 1001 |
| 7 | +  0101010111101101  =  1001001110101111  0100100111010111 | 101011010011011 | 0101010111101101 | 1000 |
| 8 | +  0101010111101101  =  1001111111000100  0100111111100010 | 010101101001101 | 0101010111101101 | 0111 |
| 9 | +  0101010111101101  =  1010010111001111  0101001011100111 | 101010110100110 | 0101010111101101 | 0110 |
| 10 | 0010100101110011 | 110101011010011 | 0101010111101101 | 0101 |
| 11 | +  0101010111101101  =  0111111101100000  0011111110110000 | 011010101101001 | 0101010111101101 | 0100 |
| 12 | +  0101010111101101  =  1001010110011101  0100101011001110 | 101101010110100 | 0101010111101101 | 0011 |
| 13 | 0010010101100111 | 010110101011010 | 0101010111101101 | 0010 |
| 14 | 0001001010110011 | 101011010101101 | 0101010111101101 | 0001 |
| 15 | +  0101010111101101  =  0110100010100000  0,0011010001010000 | 010101101010110 | 0101010111101101 | 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 |

Z1

Початок

Z2

R, W2, W3, WCT

0

X1

W1

Z3

1

Z4

1

Кінець

X2

0

ShR1,ShR2,dec

Z5

*Рисунок 2.1.4-Закодований мікроалгоритм.*

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



*Рисунок 2.1.5-Граф автомата Мура*

**2.1.8 Обробка порядків:**

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

=8; =5; =8+5=13=11012

**2.1.9 Нормалізація результату:**

Отримали результат:

Знак мантиси: 1 0 = 1.

Виконуємо зсув результату вліво, доки у першому розряді не опиниться одиниця, при цьому порядок числа зменшуємо на 1:

Mz = 0,011010001010000

0,0110100010100000← Pz=Pz-1=13-1=12=11002

Нормалізований результат:

Зн. Р Pz=11002 Зн. М Mz=0,1101000101000000

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

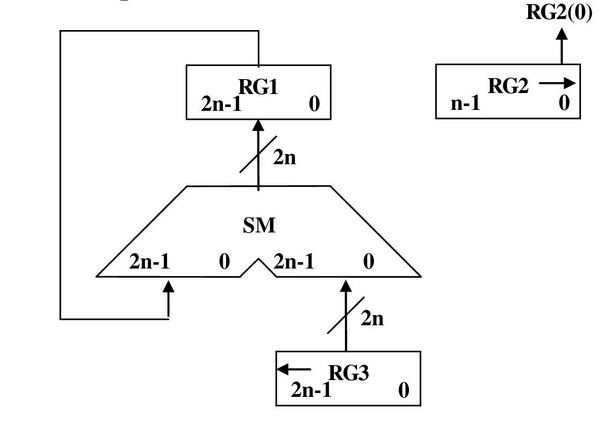
Множення мантис другим способом здійснюється з молодших розрядів, множене зсувається вліво, а сума часткових добутків залишається нерухомою.

Z=Y+ Y…+ Y;

Z=((0+ Y)+ Y)…+ Y;

Z=;

**2.2.2 Операційна схема:**



*Рисунок 2.2.1- Операційна схема.*

**2.2.3 Змістовний мікроалгоритм:**

Початок

RG2[0]

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

1

01

RG1:=RG1+RG3;

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

101

RX=0

01

Кінець

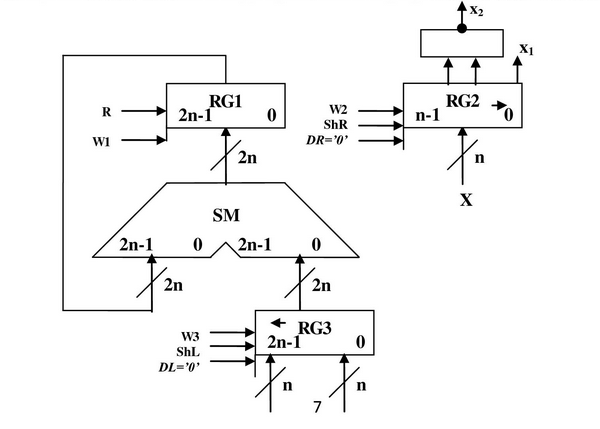
*Рисунок 2.2.2 - Змістовний мікроалгоритм.*

**2.2.4 Таблиця станів регістрів:**

*Таблиця 2.2.1-Таблиця станів регістрів.*

|  |  |  |  |
| --- | --- | --- | --- |
| № | RG1 | RG2 | RG3 |
| ПС | 0 | 100110111101110 | 000000000000000101010111101101 |
| 1 | 0 | 010011011110111 | 000000000000001010101111011010 |
| 2 | 000000000000001010101111011010 | 001001101111011 | 000000000000010101011110110100 |
| 3 | +  000000000000010101011110110100  =  000000000000100000001110001110 | 000100110111101 | 000000000000101010111101101000 |
| 4 | +  000000000000101010111101101000  =  000000000001001011001011110110 | 000010011011110 | 000000000001010101111011010000 |
| 5 | 000000000001001011001011110110 | 000001001101111 | 000000000010101011110110100000 |
| 6 | +  000000000010101011110110100000  =  000000000011110111000010010110 | 000000100110111 | 000000000101010111101101000000 |
| 7 | +  000000000101010111101101000000  =  000000001001001110101111010110 | 000000010011011 | 000000001010101111011010000000 |
| 8 | +  000000001010101111011010000000  =  000000010011111110001001010110 | 000000001001101 | 000000010101011110110100000000 |
| 9 | +  000000010101011110110100000000  =  000000101001011100111101010110 | 000000000100110 | 000000101010111101101000000000 |
| 10 | 000000101001011100111101010110 | 000000000010011 | 000001010101111011010000000000 |
| 11 | +  000001010101111011010000000000  =  000001111111011000001101010110 | 000000000001001 | 000010101011110110100000000000 |
| 12 | +  000010101011110110100000000000  =  000100101011001110101101010110 | 000000000000100 | 000101010111101101000000000000 |
| 13 | 000100101011001110101101010110 | 000000000000010 | 001010101111011010000000000000 |
| 14 | 000100101011001110101101010110 | 000000000000001 | 010101011110110100000000000000 |
| 15 | +  010101011110110100000000000000  =  0,011010001010000010101101010110 | 000000000000000 | 101010111101101000000000000000 |

**2.2.5 Функціональна схема:**



*Рисунок 2.2.3- Функціональна схема.*

**2.2.6 Закодований мікроалгоритм**

*Таблиця 2.2.2-Таблиця кодування операцій і логічних умов.*

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

Початок Z1

R,W2,W3 Z2

0

X1

1

W1 Z3

ShR, ShL Z4

0

X2

1

Кінець Z5

*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:

=810; =510; =1310=11002

**2.2.9 Нормалізація результату:**

Отримали результат:

Знак мантиси: 1 0 = 1.

Виконуємо зсув результату вліво, доки у першому розряді не опиниться одиниця, при цьому порядок числа зменшуємо на 1:

Mz = 0,011010001010000

0,0110100010100000← Pz=Pz-1=13-1=12=11002

Нормалізований результат:

Зн. Р Pz=11002 Зн. М Mz=0,1101000101000000

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

**2.3 Третій спосіб множення.**

**2.3.1Теоретичне 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.

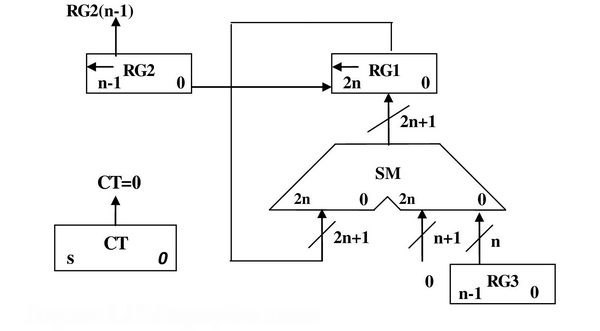
Множення мантис третім способом здійснюється зі старших розрядів множника, сума часткових добутків і множник зсуваються вліво, а множене нерухоме.

Z=Y+ Y…+ Y;

Z= Y+2(Y+2(Y…+2Y));

Z=;

**2.3.2 Операційна схема:**



*Рисунок 2.3.1 - Операційна схема*

**2.3.3 Змістовний мікроалгоритм:**

Початок

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

01

RG2[n-1]

1

RG1:=RG1+RG3;

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

01

CT=0

1

Кінець

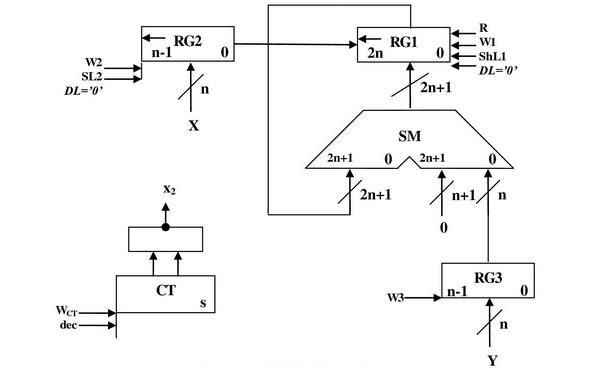
*Рисунок 2.3.2* **-** *Змістовний мікроалгоритм.*

**2.3.4 Таблиця станів регістрів:**

*Таблиця 2.3.1- Таблиця станів регістрів*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | RG1 | RG2 | RG3 | CT |
| ПС | 000000000000000000000000000000 | 100110111101110 | 101010111101101 | 1111 |
| 1 | 000000000000001010101111011010 | 001101111011100 | 101010111101101 | 1110 |
| 2 | 000000000000010101011110110100 | 011011110111000 | 101010111101101 | 1101 |
| 3 | 000000000000101010111101101000 | 110111101110000 | 101010111101101 | 1100 |
| 4 | +  000000000000000101010111101101  =  000000000000110000010101010101  000000000001100000101010101010 | 101111011100000 | 101010111101101 | 1011 |
| 5 | +  000000000000000101010111101101  =  000000000001100110000010010111  000000000011001100000100101110 | 011110111000000 | 101010111101101 | 1010 |
| 6 | 000000000110011000001001011100 | 111101110000000 | 101010111101101 | 1001 |
| 7 | +  000000000000000101010111101101  =  000000000110011101100001001001  000000001100111011000010010010 | 111011100000000 | 101010111101101 | 1000 |
| 8 | +  000000000000000101010111101101  =  000000001101000000011001111111  000000011010000000110011111110 | 110111000000000 | 101010111101101 | 0111 |
| 9 | +  000000000000000101010111101101  =  000000011010000110001011101011  000000110100001100010111010110 | 101110000000000 | 101010111101101 | 0110 |
| 10 | +  000000000000000101010111101101  =  000000110100010001101111000011  000001101000100011011110000110 | 011100000000000 | 101010111101101 | 0101 |
| 11 | 000011010001000110111100001100 | 111000000000000 | 101010111101101 | 0100 |
| 12 | +  000000000000000101010111101101  =  000011010001001100010011111001  000110100010011000100111110010 | 110000000000000 | 101010111101101 | 0011 |
| 13 | +  000000000000000101010111101101  =  000110100010011101111111011111  001101000100111011111110111110 | 100000000000000 | 101010111101101 | 0010 |
| 14 | +  000000000000000101010111101101  =  001101000101000001010110101011  011010001010000010101101010110 | 000000000000000 | 101010111101101 | 0001 |
| 15 | 110100010100000101011010101100 | 000000000000000 | 101010111101101 | 0000 |

**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 |

R, W2, W3, WCT

X1

W1

ShL1,ShL2,dec

X2

Кінець

1

0

1

0

Z2

Z3

Z4

Z5

Z1

Початок

*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:

=8; =5; =1310=11002

**2.3.9 Нормалізація результату:**

Отримали результат:

Знак мантиси: 1 0 = 1.

Виконуємо зсув результату вліво, доки у першому розряді не опиниться одиниця, при цьому порядок числа зменшуємо на 1:

Mz = 0,11010001010000

110100010100000← Pz=Pz-1=13-1=12=11002

Нормалізований результат:

Зн. Р Pz=11002 Зн. М Mz=0,1101000101000000

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

**2.4 Четвертий спосіб множення.**

**2.4.1Теоретичне 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.

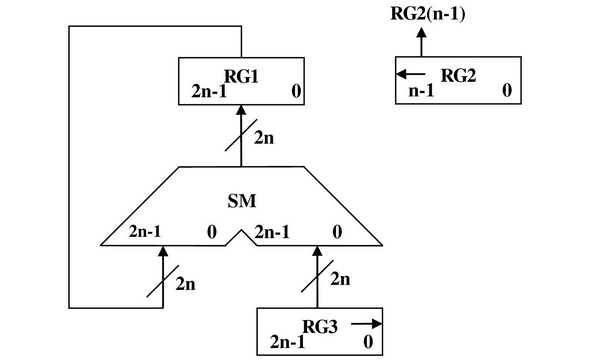
Множення здійснюється зі старших розрядів множника, сума часткових добутків залишається нерухомою, множене зсувається праворуч, множник ліворуч.

.

*.*

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

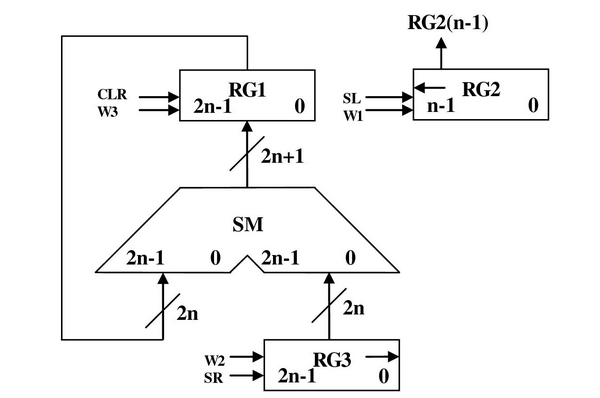
*Рисунок 2.4.2* **-** *Змістовний мікроалгоритм.*

**2.4.4 Таблиця станів регістрів:**

*Таблиця 2.4.1- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № | RG1 | RG2 | RG3 |
| ПС | 000000000000000000000000000000 | 100110111101110 | 010101011110110100000000000000 |
| 1 | 010101011110110100000000000000 | 001101111011100 | 001010101111011010000000000000 |
| 2 | 010101011110110100000000000000 | 011011110111000 | 000101010111101101000000000000 |
| 3 | 010101011110110100000000000000 | 110111101110000 | 000010101011110110100000000000 |
| 4 | +  000010101011110110100000000000  =  011000001010101010100000000000 | 101111011100000 | 000001010101111011010000000000 |
| 5 | +  000001010101111011010000000000  =  011001100000100101110000000000 | 011110111000000 | 000000101010111101101000000000 |
| 6 | 011001100000100101110000000000 | 111101110000000 | 000000010101011110110100000000 |
| 7 | +  000000010101011110110100000000  =  011001110110000100100100000000 | 111011100000000 | 000000001010101111011010000000 |
| 8 | +  000000001010101111011010000000  =  011010000000110011111110000000 | 110111000000000 | 000000000101010111101101000000 |
| 9 | +  000000000101010111101101000000  =  011010000110001011101011000000 | 101110000000000 | 000000000010101011110110100000 |
| 10 | +  000000000010101011110110100000  =  011010001000110111100001100000 | 011100000000000 | 000000000001010101111011010000 |
| 11 | 011010001000110111100001100000 | 111000000000000 | 000000000000101010111101101000 |
| 12 | +  000000000000101010111101101000  =  011010001001100010011111001000 | 110000000000000 | 000000000000010101011110110100 |
| 13 | +  000000000000010101011110110100  =  011010001001110111111101111100 | 100000000000000 | 000000000000001010101111011010 |
| 14 | +  000000000000001010101111011010  =  011010001010000010101101010110 | 000000000000000 | 000000000000000101010111101101 |
| 15 | 011010001010000010101101010110 | 000000000000000 | 000000000000000010101011110110 |

**2.4.5Функціональна схема:**



*Рисунок 2.4.3* **-** *Функціональна схема.*

**2.4.6 Закодований мікроалгоритм**

*Таблиця 2.4.2-Таблиця кодування операцій і логічних умов.*

|  |  |  |  |
| --- | --- | --- | --- |
| Кодування мікрооперацій | | Кодування логічних умов | |
| МО | УС | ЛУ | Позначення |
| RG1:=0  RG2:=X  RG3:=Y RG1:=RG1+RG3  RG3:=0.r(RG3) RG2:=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.*

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



*Рисунок 2.4.5 - Граф автомата Мура*

**2.4.8 Обробка порядків:**

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

=; =; ==

**2.4.9 Нормалізація результату:**

Отримали результат:

Знак мантиси: 1 0 = 1.

Виконуємо зсув результату вліво, доки у першому розряді не опиниться одиниця, при цьому порядок числа зменшуємо на 1:

Mz = 0,011010001010000

0,0110100010100000← Pz=Pz-1=13-1=12=11002

Нормалізований результат:

Зн. Р Pz=11002 Зн. М Mz=0,1101000101000000

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

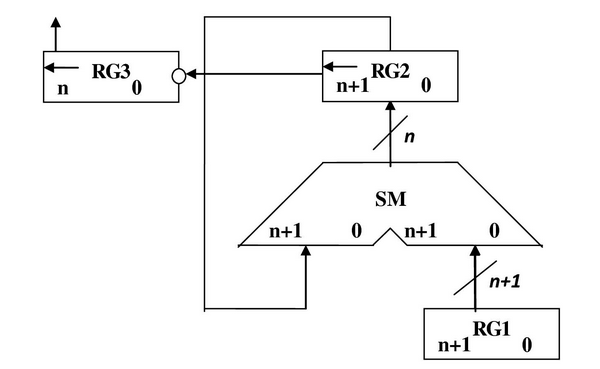
**2.5. Першиий спосіб ділення.**

**2.5.1Теоретичне обґрунтування першого способу ділення:**

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

При реалізації ділення за першим методом здійснюється зсув вліво залишку при нерухомому дільнику. Черговий залишок формується в регістрі Р2 (у вихідному стані 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:**

Початок

RG3:=0

RG2:=X

RG1:=Y

RG2[n+1]

Кінець

RG3:=l(RG3).

RG2:=l(RG2).0

RG2:=RG2+RG1

RG2:=RG2++1

RG2[n+1]

0

1

1

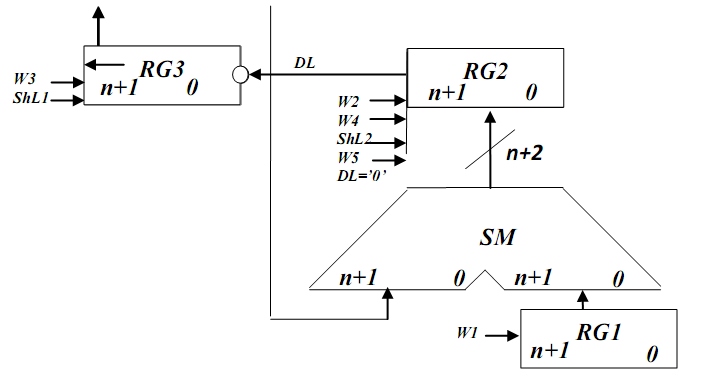
0

*Рисунок 2.5.2-Змістовний мікроалгоритм*

**2.5.4 Таблиця станів регістрів:**

|  |  |  |  |
| --- | --- | --- | --- |
| № | RG3(Z) | RG2(X) | RG1(Y) |
| ПС | 0000000000000000 | 00100110111101110 | 101010111101101 |
| 1 | 0000000000000001 | 01001101111011100  +  11010101000010011  =  00100010111101111 | 101010111101101 |
| 2 | 0000000000000011 | 01000101111011110  +  11010101000010011  =  00011010111110001 | 101010111101101 |
| 3 | 0000000000000111 | 00110101111100010  +  11010101000010011  =  00001010111110101 | 101010111101101 |
| 4 | 0000000000001111 | 00010101111101010  +  11010101000010011  =  11101010111111101 | 101010111101101 |
| 5 | 0000000000011110 | 11010101111111010  +  00101010111101101  =  00000000111100111 | 101010111101101 |
| 6 | 0000000000111101 | 00000001111001110  +  11010101000010011  =  11010110111100001 | 101010111101101 |
| 7 | 0000000001111010 | 10101101111000010  +  00101010111101101  =  11011000110101111 | 101010111101101 |
| 8 | 0000000011110100 | 10110001101011110  +  00101010111101101  =  11011100101001011 | 101010111101101 |
| 9 | 0000000111101000 | 10111001010010110  +  00101010111101101  =  11100100010000011 | 101010111101101 |
| 10 | 0000001111010000 | 11001000100000110  +  00101010111101101  =  11110011011110011 | 101010111101101 |
| 11 | 0000011110100000 | 11100110111100110  +  00101010111101101  =  00010001111010011 | 101010111101101 |
| 12 | 0000111101000001 | 00100011110100110  +  11010101000010011  =  11111000110111001 | 101010111101101 |
| 13 | 0001111010000010 | 11110001101110010  +  00101010111101101  =  00011100101011111 | 101010111101101 |
| 14 | 0011110100000101 | 00111001010111110  +  11010101000010011  =  00001110011010001 | 101010111101101 |
| 15 | 0111101000001011 | 00011100110100010  +  11010101000010011  =  11110001110110101 | 101010111101101 |
| 16 | 1111010000010110 | 11100011101101010  +  00101010111101101  =  00001110101010111 | 101010111101101 |

**2.5.5 Функціональна схема:**



*Рисунок 2.5.3 – Функціональна схема*

**2.5.6 Закодований мікроалгоритм**

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

|  |  |  |  |
| --- | --- | --- | --- |
| Кодування мікрооперацій | | Кодування логічних умов | |
| МО | УС | ЛУ | Позначення |
| RG3:=0  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 |

Z1

Початок

Z2

W3, W2, W1

Z33

Кінець

X2

W5

X1

W4

ShL1, ShL2

1

0

Z5

Z4

1

0

Z6

*Рисунок 2.5.4-Закодований мікроалгоритм.*

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



*Рисунок 2.5.5 - Граф управляючого автомата.*

**2.5.8 Обробка порядків:**

Порядок частки дорівнює:

У даному випадку =810; =510; =310=0112;

**2.5.8 Нормалізація результату:**

Отримали результат:

Знак мантиси: 1 0 = 1.

Нормалізація мантиси не потрібна.

Mz=11110100000101102;

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

**2.6. Другий спосіб ділення.**

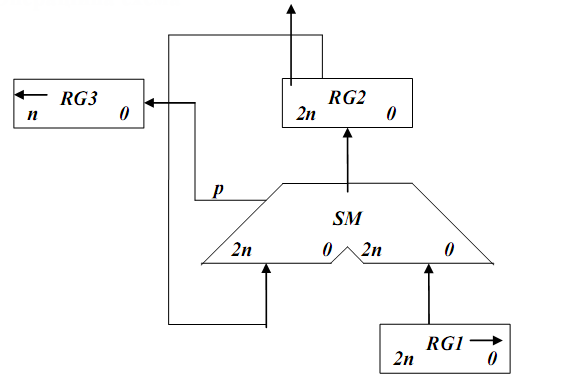
**2.6.1Теоретичне обгрунтування другого способу ділення:**

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

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

прискорення відносно 1-го способу.

**2.6.2 Операційна схема**



*Рисунок 2.6.1-Операційна схема*

**2.6.3 Змістовний мікроалгоритм**

Початок

RG3:=0

RG1:=Y

RG2=X

RG2[2n+1]

Кінець

RG2:=RG2+RG1

RG1:=0.r(RG1)

RG3:=l(RG3).SM(p)

RG2:=RG2++1

RG1:=0.r(RG1)

RG3:=l.(RG3).SM(p)

RG3[n]

0

1

0

1

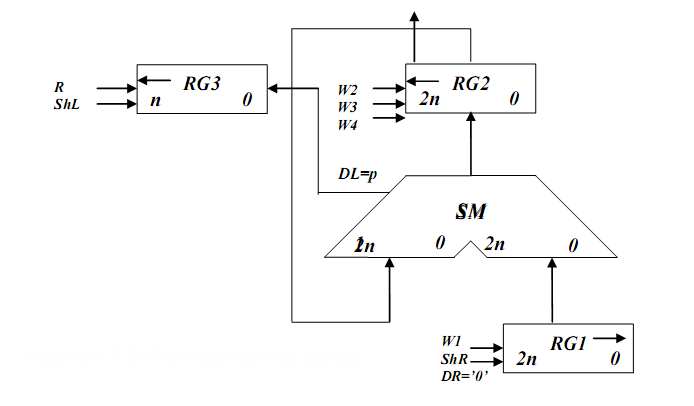
*Рисунок 2.6.2-Змістовний мікроалгоритм*

**2.6.4 Таблиця станів регістрів**

*Таблиця 2.6.1- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № | RG3(Z) | RG2(X) | RG1(Y) |
| ПС | 0000000000000001 | 010011011110111000000000000000 | 001010101111011010000000000000 |
| 1 | 0000000000000011 | 010011011110111000000000000000  +  110101010000100110000000000000  =  001000101111011110000000000000 | 000101010111101101000000000000 |
| 2 | 0000000000000111 | 001000101111011110000000000000  +  111010101000010011000000000000  =  000011010111110001000000000000 | 000010101011110110100000000000 |
| 3 | 0000000000001111 | 000011010111110001000000000000  +  111101010100001001100000000000  =  000000101011111010100000000000 | 000001010101111011010000000000 |
| 4 | 0000000000011110 | 000000101011111010100000000000  +  111110101010000100110000000000  =  111111010101111111010000000000 | 000000101010111101101000000000 |
| 5 | 0000000000111101 | 111111010101111111010000000000  +  000000101010111101101000000000  =  000000000000111100111000000000 | 000000010101011110110100000000 |
| 6 | 0000000001111010 | 000000000000111100111000000000  +  111111101010100001001100000000  =  111111101011011110000100000000 | 000000001010101111011010000000 |
| 7 | 0000000011110100 | 111111101011011110000100000000  +  000000001010101111011010000000  =  111111110110001101011110000000 | 000000000101010111101101000000 |
| 8 | 0000000111101000 | 111111110110001101011110000000  +  000000000101010111101101000000  =  111111111011100101001011000000 | 000000000010101011110110100000 |
| 9 | 0000001111010000 | 111111111011100101001011000000  +  000000000010101011110110100000  =  111111111110010001000001100000 | 000000000001010101111011010000 |
| 10 | 0000011110100000 | 111111111110010001000001100000  +  000000000001010101111011010000  =  111111111111100110111100110000 | 000000000000101010111101101000 |
| 11 | 0000111101000001 | 111111111111100110111100110000  +  000000000000101010111101101000  =  000000000000010001111010011000 | 000000000000010101011110110100 |
| 12 | 0001111010000010 | 000000000000010001111010011000  +  111111111111101010100001001100  =  111111111111111100011011100100 | 000000000000001010101111011010 |
| 13 | 0011110100000101 | 111111111111111100011011100100  +  000000000000001010101111011010  =  000000000000000111001010111110 | 000000000000000101010111101101 |
| 14 | 0111101000001011 | 000000000000000111001010111110  +  111111111111111010101000010011  =  000000000000000001110011010001 | 000000000000000010101011110110 |
| 15 | 1111010000010110 | 000000000000000001110011010001  +  111111111111111101010100001010  =  111111111111111111000111011011 | 000000000000000001010101111011 |

**2.6.5 Функціональна схема з відображенням управляючих сигналів**



*Рисунок 2.6.3-Функціональна схема*

**2.6.6 Закодований мікроалгоритм**

*Таблиця 2.6.2- Таблиця кодування мікрооперацій*

|  |  |
| --- | --- |
| Таблиця кодування мікрооперацій | |
| МО | УС |
| RG3:=0  RG1:=Y  RG2:=X  RG2:=RG2+RG1  RG1:=0.r(RG1)  RG3:=l(RG3).SM(p)  RG2:==RG2++1 | R  W1  W2  W3  ShR  ShL  W4 |

|  |  |
| --- | --- |
| Таблиця кодування логічних умов | |
| ЛУ | Позначення |
| RG2[2n+1]  RG3[n] | X1  X2 |

Z1

Початок

R, W1, W2

X1

Кінець

W4, ShR, ShL

W3, ShR, ShL

X2

Z2

1

0

Z4

Z3

0

1

Z5

*Рисунок 2.6.4- Закодований мікроалгоритм*

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



*Рисунок 2.6.5- Граф автомата Мура*

**2.6.8 Обробка порядків:**

Порядок частки буде дорівнювати:

В моєму випадку =810; =510; =32=0112;

**2.6.9 Нормалізація результату:**

Отримали результат:

Знак мантиси: 1 0 = 1.

Нормалізація мантиси не потрібна.

Mz=11110100000101102

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

**2.7. Операція додавання чисел. Операція віднімання чисел.**

**2.7.1 Теоретичне обґрунтування способу**

В пам’яті числа зберігаються у ПК. На першому етапі додавання чисел з плаваючою комою виконують вирівнювання порядків до числа із старшим порядком. На другому етапі виконують додавання мантис. Додавання мантис виконується у 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 1 розряд вліво і на n розрядів вправо.

Для отримання різниці чисел знак мантиси від’ємника змінюється на протилежний.

1. Порівняння порядків.

Px=+810=+10002

Py=+510=+01012

810-510=310=112

Pz=1011

2. Вирівнювання порядків.

Робимо зсув вправо мантиси числа Y, зменшуючи на кожному кроці, доки не стане 0.

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

|  |  |  |
| --- | --- | --- |
| MY | ∆ | Мікрооперація |
| 0,101010111101101 | 11 | Початковий стан |
| 0,010101011110110 | 10 | My= 0.r(My); ∆=∆-1 |
| 0, 001010101111011 | 01 | My= 0.r(My); ∆=∆-1 |
| 0, 000101010111101 | 00 | My= 0.r(My); ∆=∆-1 |

3.1 Додавання мантис у модифікованому ДК.

X мдк = 11.+1 =11.011001000010001+1 = 11.011001000010010

Yмдк = 00. 000101010111101

*Таблиця 2.7.2-Додавання мантис у модифікованому ДК*

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

3.2 Віднімання мантис у модифікованому ДК.

X мдк = 1. +1 = 11.011001000010001+1 = 11.011001000010010

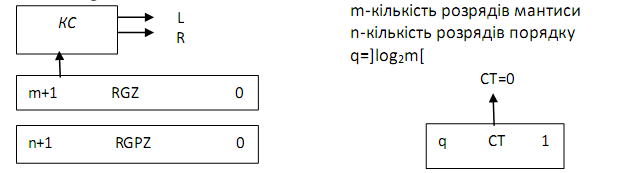
-Yмдк =1.  +1 = 11. 111010101000010+1 = 11. 111010101000011

*Таблиця 2.7.3-Віднімання мантис у модифікованому ДК*

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

4. Нормалізація результату (в модифікованому прямому коді).

**2.7.2 Операційна схема**



*Рисунок 2.7.1-Операційна схема*

Виконаємо синтез КС для визначення порушення нормалізації.

z'0 – старший знаковий розряд мантиси,

z0 – молодший знаковий розряд мантиси,

z1 – старший розряд мантиси.

L та R є функціями порушення нормалізації мантиси вліво та вправо відповідно.

*Таблиця 2.7.3-Визначення порушення нормалізації*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Розряди регістру  RGZ | | | Значення  функцій | |
| z'0 | z0 | z1 | L | R |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |



Знак мантиси результату встановлюємо за Z'0 до нормалізації.

**2.7.5 Змістовний алгоритм**

Початок

CT:=m;

RGZ:=Z;

Кінець

RGZ:=l(RGZ).0

RGPZ:=RGPZ-1

CT:=CT-1

RGZ:=0.r(RGZ)

RGZP:=RGZP+1



CT=0

1



0

1



0

0

1

RGZ:=minM

RGPZ:=minP

*Рисунок 2.7.2-Змістовний мікроалгоритм*

**2.7.4 Таблиця станів регістрів**

1) Нормалізація результату додавання.

*Таблиця 2.7.5- Таблиця станів регістрів*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№**  **такту** | **RGPZ** | **RGZ** | **ЛПН(L)** | **ППН(R)** | **СT** | **Мікрооперація** |
| **ПС** | 0001000 | 011110011001111 | 0 | 0 | 100 |  |

2) Нормалізація результату віднімання.

*Таблиця 2.7.6- Таблиця станів регістрів*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№**  **такту** | **RGPZ** | **RGZ** | **ЛПН(L)** | **ППН(R)** | **СT** | **Мікрооперація** |
| **ПС** | 0001000 | 000011101010101 | 0 | 0 | 100 |  |

**2.7.5 Функціональна схема з відображенням керуючих сигналів**



*Рисунок 2.7.3 – Функціональна схема*

**2.7.6 Закодований мікроалгоритм**

*Таблиця 2.7.7 – Таблиця кодування*

|  |  |
| --- | --- |
| Таблиця кодування мікрооперацій | |
| МО | УС |
| CT:=m  RGZ:=Z  RGZ:=RGZ(m+2).r(RGZ)  RGPZ:=RGPZ+1  RGZ:=l(RGZ).0  RGPZ:=RGPZ-1  CT:=CT-1  RGZ:=minM  RGPZ:=minP | W  W1  ShR  Inc  ShL  Dec  Dec  W2  W3 |

|  |  |
| --- | --- |
| Таблиця кодування логічних умов | |
| ЛУ | Позначення |
| CT=0 | X1  X2  X3 |

Z2

Z1

Початок

W, W1

X1

Кінець

ShL, dec

ShR, Inc

X2

X3



1

Z3

0

0

1

Z4

0

1

Z5

W2, W3

Z6

*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**

1) PX+Y= 810 =10002

2) PX-Y = 810 = 10002

**2.7.9 Форма запису результату з плаваючою комою**

1) Результат додавання Z=X+Y.

Mz(мпк) = 11.100001100110001

Pz = 810 =10002 Mz = 1000011001100012

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

2) Результат віднімання Z=X-Y.

Mz(мпк) = 11.111100010101011

Pz = 810 =10002 Mz = 1111000101010112

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

**Завдання 3**

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

x3 x2 x1 = 1102 = 610.

**3.1 Таблиця кодування сигналів**

*Таблиця 3.1 – Таблиця кодування сигналів*

|  |  |
| --- | --- |
| R, W1, W2 | Y1 |
| ShR, ShL | Y2 |
| W3 | Y3 |
| W4 | Y4 |

**3.2 Мікроалгоритм в термінах управляючого автомата**

Z1

Початок

Y1

X1

Кінець

Y4, Y2

Y3, Y2

X2

Z2

1

0

Z4

Z3

0

1

Z1

*Рисунок 3.1 – Закодований мікроалгоритм*

**3.3 Граф упарвляючого автомата**



*Рисунок 3.2 – Граф циклічного автомата*

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

*Таблиця 3.2 – Таблиця переходів*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Перехід | Старий  стан | | Новий  стан | | Вхідні  сигнали | | Вихідні сигнали | | | | Функції збудження тригерів | |
| Q2t | Q1t | Q2t+1 | Q1t+1 | X2 | X1 | Y4 | Y3 | Y2 | Y1 | D2 | D1 |
| Z1-Z2 | 0 | 0 | 0 | 1 | - | - | 0 | 0 | 0 | 0 | 0 | 1 |
| Z2-Z3 | 0 | 1 | 1 | 0 | - | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| Z2-Z4 | 0 | 1 | 1 | 1 | - | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| Z3-Z4 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| Z3-Z1 | 1 | 0 | 0 | 0 | 1 | - | 0 | 1 | 1 | 0 | 0 | 0 |
| Z3-Z3 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| Z4-Z3 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| Z4-Z1 | 1 | 1 | 0 | 0 | 1 | - | 1 | 0 | 1 | 0 | 0 | 0 |
| Z4-Z4 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |

**3.5 Функцій збудження тригерів. Мінімізація функцій методом діаграм Вейча.**

****

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| D2 |  | Q1 | |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Q2 |  | 1 | 0 | 0 | 1 |  |  |
|  | 1 | 0 | 0 | 1 |  | X1 |
|  |  | 1 | 1 | 0 | 0 |  |
|  |  | 1 | 1 | 0 | 0 |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  | X2 | |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| D1 |  | Q1 | |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Q2 |  | 0 | 0 | 0 | 0 |  |  |
|  | 1 | 0 | 0 | 1 |  | X1 |
|  |  | 1 | 1 | 1 | 1 |  |
|  |  | 0 | 0 | 1 | 1 |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  | X2 | |  |  |  |

*Рисунок 3.3 – Мінімізація функцій тригерів*

****

**3.6 Функції виходів управляючого автомата.**

****