SM

n + 2

n + 2

n + 2

n + 1

n + 2

1

1

1

1

1 + 1

Rx

Rz

Ry

*Рисунок 2.21 – Операційна схема*

c

2.5.3) *Змістовний мікроалгоритм:*

так

ні

1

0

Початок

Rz := 0..01М;

Rx := X2;

Ry := Y2;

Rx[1]

Rx := Rx + Ry;

Rx := l(Rx).0;

Rz := l(Rx).[1];

Rz[1]

Кінець

*Рисунок 2.22 – Змістовний мікроалгоритм*

Rx := Rx + + c;

Перед початком ділення ми зробимо зсув праворуч діленого і додамо 1 в розряд, адже мантиса X мусить бути меншою мантиси Y. Це означає, що можливе порушення нормалізації вправо на один розряд.

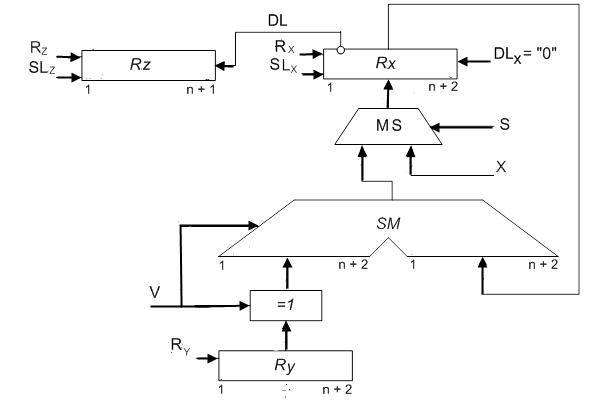
2.5.4) *Таблиця станів регістрів (лічильника):*

*Таблиця 2.13 – Таблиця станів регістрів*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Rz | Rx | Ry | Коментар |
| П.С. | 000000000000000**1** | **0**0100101110100100 | 00101001110100001 | Y2 ДК = 11010110001011111 |
| 1 | 00000000000000**1**0 | 00100101110100100  +11010110001011111  11111100000000011  **1**1111000000000110 |  | +  ⮘ |
| 2 | 0000000000000**1**01 | 11111000000000110  +00101001110100001  00100001110100111  **0**1000011101001110 |  | +  ⮘ |
| 3 | 000000000000**1**011 | 01000011101001110  +11010110001011111  00011001110101101  **0**0110011101011010 |  | +  ⮘ |
| 4 | 00000000000**1**0111 | 00110011101011010  +11010110001011111  00001001110111001  **0**0010011101110010 |  | +  ⮘ |
| 5 | 0000000000**1**01110 | 00010011101110010  +11010110001011111  11101001111010001  **1**1010011110100010 |  | +  ⮘ |
| 6 | 000000000**1**011100 | 11010011110100010  +00101001110100001  11111101101000011  **1**1111011010000110 |  | +  ⮘ |
| 7 | 00000000**1**0111001 | 11111011010000110  +00101001110100001  00100101000100111  **0**1001010001001110 |  | +  ⮘ |
| 8 | 0000000**1**01110011 | 01001010001001110  +11010110001011111  00100000010101101  **0**1000000101011010 |  | +  ⮘ |
| 9 | 000000**1**011100111 | 01000000101011010  +11010110001011111  00010110110111001  **0**0101101101110010 |  | +  ⮘ |
| 10 | 00000**1**0111001111 | 00101101101110010  +11010110001011111  00000011111010001  **0**0000111110100010 |  | +  ⮘ |
| 11 | 0000**1**01110011110 | 00000111110100010  +11010110001011111  11011110000000001  **1**0111100000000010 |  | +  ⮘ |
| 12 | 000**1**011100111100 | 10111100000000010  +00101001110100001  11100101110100011  **1**1001011101000110 |  | +  ⮘ |
| 13 | 00**1**0111001111000 | 11001011101000110  +00101001110100001  11110101011100111  **1**1101010111001110 |  | +  ⮘ |
| 14 | 0**1**01110011110001 | 11101010111001110  +00101001110100001  00010100101101111  **0**0101001011011110 |  | +  ⮘ |
| 15 | **1011100111100010** | 00101001011011110  +11010110001011111  11111111100111101  **1**1111111001111010 |  | +  ⮘ |

Результат ділення виділено в таблиці.

2.5.5) *Функціональна схема з відображенням управляючих сигналів:*

**

*Рисунок 2.23 – Функціональна схема*

2.5.6) *Закодований мікроалгоритм:*

*Таблиця 2.14 – Кодування мікрооперацій*

|  |  |  |
| --- | --- | --- |
| Мікрооперації | | Управляючі сигнали |
| Rx := X2 | | RX |
| Ry := Y2 | | RY |
| Rz := 0..01М | | RZ |
| Rx := Rx + Ry | | Sum |
| Rx := Rx + + C | | V |
| Rx := l(Rx).0 | | SLX |
| Rz := l(Rz). | | SLZ |
|  | | |
| Логічна умова | Позначення | |
| Rx[1] | X1 | |
| Rz[1] | X2 | |

*Таблиця 2.15 – Кодування логічних умов*

*Таблиця 2.3 – Кодування логічних умов*

*Таблиця 2.3 – Кодування логічних умов*

*Таблиця 2.3 – Кодування логічних умов*

*Рисунок 2.24 – Кодований мікроалгоритм*

RX, RY, RZ, S

Z1

Z2

Z4

Z5

Z6

так

ні

0

1

Початок

X1

V

SLX, SLZ

X2

Кінець

Z3

Sum

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

–

*Рисунок 2.25 – Граф управляючого автомата*

, 

–

, 



X1



–

Q3Q2Q1

000

001

011

010

110

111

2.5.8) *Обробка порядків:*

PZ = PX – PY.

Отже порядок результату PZ = 910 – 510 = 410 = 01002.

Порушення нормалізації, тому виконаємо зсув мантиси ліворуч на розряд та віднімання 1 від порядку PZ = 410 – 110 = 310 = 0112.

MZ = 0111001111000102.

= 1110011110001002.

Знак мантиси .

2.5.9) *Форма запису нормалізованого результату з плаваючою комою в пам’ять:*

Запишемо результат ділення Z2 в розрядну сітку в прямому коді у вигляді ПОРЯДОК МАНТИСА:

Z2 : **0.** 0 0 0 0 0 1 1 **1,** 1 1 1 0 0 1 1 1 1 0 0 0 1 0 0

Знакові розряди виділені.

**2.6) Другий спосіб ділення (з зсувом дільника) Z = X/Y**

2.6.1) *Теоретичне обґрунтування способу*: нехай ділене Х і дільник Y є n-розрядними правильними дробами, поданими в прямому коді. При діленні чисел окремо визначається порядок результату і мантиса. Потрібно враховувати, що мантиса Х повинна бути меншою за мантису Y. Для нормалізації мантису Х зсувають на один розряд праворуч, а порядок збільшують на 1. Знак результату визначається шляхом підсумовування по модулю 2 цифр, записаних в знакових розрядах (операція ).

В другому способі ділення при реалізації здійснюється зсув вправо дільника при нерухомій остачі. Цей спосіб забезпечує кращу швидкодію, але є складнішим.

2.6.2) *Операційна схема:*

p

SM

2n + 1

2n + 1

2n + 1

n + 2

2n + 1

1

1

1

1

1 + 1

Rx

Rz

Ry

*Рисунок 2.26 – Операційна схема*

c

2.6.3) *Змістовний мікроалгоритм:*

так

ні

1

0

Початок

Rz := 0..01М;

Rx := 0.X20..0;

Ry := 0.Y20..0;

Rx[1]

Rx := Rx + Ry;

Rz := l(Rx).p;

Ry := 0.r(Ry);

Rz[1]

Кінець

*Рисунок 2.27 – Змістовний мікроалгоритм*

Rx := Rx + + c;

Перед початком ділення ми зробимо зсув праворуч діленого і додамо 1 в розряд, адже мантиса X мусить бути меншою мантиси Y. Це означає, що можливе порушення нормалізації вправо на один розряд.

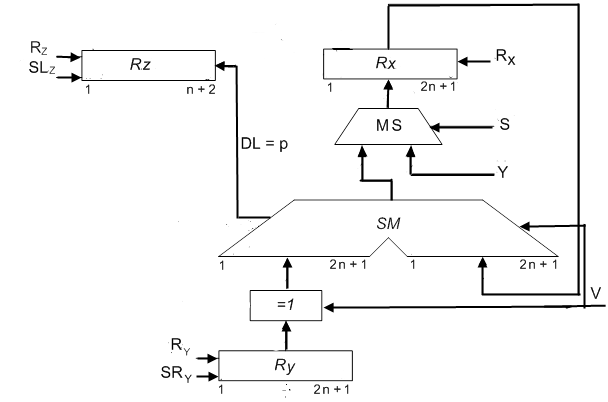
2.6.4) *Таблиця станів регістрів (лічильника):*

*Таблиця 2.16 – Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № | Rz | Rx | Ry |
| П.С. | 0000000000000000**1** | **0**100101110100100000000000000000 | 0010100111010000100000000000000ПК  1101011000101111100000000000000ДК |
| 1 | 000000000000000**1**0 | 0100101110100100000000000000000  +1101011000101111100000000000000  **0**010000111010011100000000000000 | 0001010011101000010000000000000ПК  1110101100010111110000000000000ДК |
| 2 | 00000000000000**1**01 | 0010000111010011100000000000000  +1110101100010111110000000000000  **0**000110011101011010000000000000 | 0000101001110100001000000000000ПК  1111010110001011111000000000000ДК |
| 3 | 0000000000000**1**011 | 0000110011101011010000000000000  +1111010110001011111000000000000  **0**000001001110111001000000000000 | 0000010100111010000100000000000ПК  1111101011000101111100000000000ДК |
| 4 | 000000000000**1**0111 | 0000001001110111001000000000000  +1111101011000101111100000000000  **1**111110100111101000100000000000 | 0000001010011101000010000000000ПК  1111110101100010111110000000000ДК |
| 5 | 00000000000**1**01110 | 1111110100111101000100000000000  +0000001010011101000010000000000  **1**111111111011010000110000000000 | 0000000101001110100001000000000ПК  1111111010110001011111000000000ДК |
| 6 | 0000000000**1**011100 | 1111111111011010000110000000000  +0000000101001110100001000000000  **0**000000100101000100111000000000 | 0000000010100111010000100000000ПК  1111111101011000101111100000000ДК |
| 7 | 000000000**1**0111001 | 0000000100101000100111000000000  +1111111101011000101111100000000  **0**000000010000001010110100000000 | 0000000001010011101000010000000ПК  1111111110101100010111110000000ДК |
| 8 | 00000000**1**01110011 | 0000000010000001010110100000000  +1111111110101100010111110000000  **0**000000000101101101110010000000 | 0000000000101001110100001000000ПК  1111111111010110001011111000000ДК |
| 9 | 0000000**1**011100111 | 0000000000101101101110010000000  +1111111111010110001011111000000  **0**000000000000011111010001000000 | 0000000000010100111010000100000ПК  1111111111101011000101111100000ДК |
| 10 | 000000**1**0111001111 | 0000000000000011111010001000000  +1111111111101011000101111100000  **1**111111111101111000000000100000 | 0000000000001010011101000010000ПК  1111111111110101100010111110000ДК |
| 11 | 00000**1**01110011110 | 1111111111101111000000000100000  +0000000000001010011101000010000  **1**111111111111001011101000110000 | 0000000000000101001110100001000ПК  1111111111111010110001011111000ДК |
| 12 | 0000**1**011100111100 | 1111111111111001011101000110000  +0000000000000101001110100001000  **1**111111111111110101011100111000 | 0000000000000010100111010000100ПК  1111111111111101011000101111100ДК |
| 13 | 000**1**0111001111000 | **1**111111111111110101011100111000  +0000000000000010100111010000100  **0**000000000000001010010110111100 | 0000000000000001010011101000010ПК  1111111111111110101100010111110ДК |
| 14 | 00**1**01110011110001 | **0**000000000000001010010110111100  +1111111111111110101100010111110  **1**111111111111111111111001111010 | 0000000000000000101001110100001ПК  1111111111111111010110001011111ДК |
| 15 | 0**1**011100111100010 | 1111111111111111111111001111010  +1111111111111111010110001011111  **1**111111111111111010101010011001 | 0000000000000000010100111010000ПК  1111111111111111101011000110000ДК |
| 16 | **10111001111000101** | 1111111111111111010101010011001  +1111111111111111101011000110000  1111111111111111000000011001001 | 0000000000000000001010011101000ПК  1111111111111111110101100011000ДК |

Результат ділення виділено в таблиці.

2.6.5) *Функціональна схема з відображенням управляючих сигналів:*

**

*Рисунок 2.28 – Функціональна схема*

2.6.6) *Закодований мікроалгоритм:*

*Таблиця 2.17 – Кодування мікрооперацій*

|  |  |  |
| --- | --- | --- |
| Мікрооперації | | Управляючі сигнали |
| Rx := 0.X20..0 | | RX |
| Ry := 0.Y20..0 | | RY |
| Rz := 0..01М | | RZ |
| Rx := Rx + Ry | | Sum |
| Rx := Rx + + C | | V |
| Rz := l(Rz).p | | SLZ |
| Ry := 0.r(Ry) | | SRY |
|  | | |
| Логічна умова | Позначення | |
| Rx[1] | X1 | |
| Rz[1] | X2 | |

*Таблиця 2.18 – Кодування логічних умов*

*Таблиця 2.3 – Кодування логічних умов*

*Таблиця 2.3 – Кодування логічних умов*

*Таблиця 2.3 – Кодування логічних умов*

*Рисунок 2.29 – Кодований мікроалгоритм*

RX, RY, RZ, S

Z1

Z2

Z4

Z5

Z6

так

ні

0

1

Початок

X1

V

SLZ, SRY

X2

Кінець

Z3

Sum

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

–

*Рисунок 2.30 – Граф управляючого автомата*

, 

–

, 



X1



–

Q3Q2Q1

000

001

011

010

110

111

2.6.8) *Обробка порядків:*

PZ = PX – PY.

Отже порядок результату PZ = 910 – 510 = 410 = 01002.

Порушення нормалізації, тому виконаємо зсув мантиси ліворуч на розряд та віднімання 1 від порядку PZ = 410 – 110 = 310 = 0112.

MZ = 0111001111000102.

= 1110011110001002.

Знак мантиси .

2.6.9) *Форма запису нормалізованого результату з плаваючою комою в пам’ять:*

Запишемо результат ділення Z2 в розрядну сітку в прямому коді у вигляді ПОРЯДОК МАНТИСА:

Z2 : **0.** 0 0 0 0 0 1 1 **1,** 1 1 1 0 0 1 1 1 1 0 0 0 1 0 0

Знакові розряди виділені.

**2.7) Додавання Z = X + Y**

2.7.1) *Теоретичне обґрунтування способу*: перш за все необхідно зрівняти порядки чисел, інакше правильне додавання мантис неможливе.