Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

# Комп’ютерна арифметика

# Розрахункова робота по курсу «КЛ-2»

Виконав: студент групи ІО-92

Железняк Олександр Юрійович

Керівник: Жабін В.І.

2010 р.

**Обґрунтування варіанту**

Номер залікової книжки: 920610 = 10 001 111 110 1102

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Двійково-десяткове число | | | | Двійкове число |
| 1000 | 100 | 10 | 1 |
| 1001 | 0010 | 0000 | 0110 |  |
| 100 | 1001  +  1101  0110 | 0000 | 0011 | 0 |
| 10 | 0011 | 0000 | 0001 | 10 |
| 1 | 0001 | 1000  +  1101  0101 | 0000 | 110 |
|  | 1000  +  1101  0101 | 1010  +  1101  0111 | 1000  +  1101  0101 | 0110 |
|  | 010 | 1011  +  1101  1000 | 1010  +  1101  0111 | 10110 |
|  | 01 | 0100 | 0011 | 110110 |
|  | 0 | 1010  +  1101  0111 | 0001 | 1110110 |
|  |  | 0011 | 1000  +  1101  0101 | 11110110 |
|  |  | 001 | 1010  +  1101  0101 | 111110110 |
|  |  | 00 | 1011  +  1101  1000 | 1111110110 |
|  |  | 0 | 0100 | 01111110110 |
|  |  |  | 0010 | 001111110110 |
|  |  |  | 001 | 0001111110110 |
|  |  |  | 00 | 10001111110110 |

X2 = –11111111,1001110

Y2 = +11111,1111001101

**Виконання роботи**

***Завдання 1***

В прямому коді:

**X2:**

Px=0.0001000 Ex=10001000 Mx = 1.111111111001110

### Y2:

Py=0.0000101 Ey=10000101 My = 0.111111111001101

***Завдання 2***

**1. Операція множення чисел**.

Z=XY, де X-множене, Y-множник.

**1.1 Перший спосіб множення.**

**1.1.1 Теоретичне обгрунтування.**

Під час множення першим способом в першому такті і-го циклу аналізується значення RG2, в якому знаходиться чергова цифра множника. Вміст RG3 додається до суми часткових добутків, що знаходяться в регістрі RG1, якщо RG2(1)=1, або не додається, якщо RG2(1)=0. В другому такті здійснюється правий зсув в регістрах RG1 і RG2, що еквівалентно множенню їх вмісту на 2-1. При зсуві цифра молодшого розряду регістру RG1 записується у вивільнюваний старший розряд регістру RG2. Після виконання n циклів молодші розряди 2n-розрядного добутку будуть записані в регістр RG2, а старші – у RG1.

Час множення, якщо не застосовувати методи прискорення операції, визначається виразом tм=n(tп+tз), де tп і tз – тривалості тактів підсумування і зсуву відповідно.

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:

що рівносильно

.

Отже, сума часткових добутків в *i*-му циклі, де , зводиться до обчислення виразу:

.

**1.1.2 Операційна схема.**

***RG2(1)***

***RG2***

***RG1***

***1***

***n+1***

***1***

***n***

***n+1***

***n+1***

***n+1***

***1***

***1***

***SM***

***n+1***

***n+1***

***1***

***RG3***

***n+1***

***CT***

***1***

***q***

***CT=0***

*Рисунок 1.1- Операційна схема.*

**1.1.3 Змістовний мікроалгоритм.**

Початок

Кінець

RG1:=0

RG2:=X

RG3:=Y

CT:=15

RG1:=RG1+RG3

RG1:=0.r(RG1)

RG2(n):=RG1(1).r(RG2)

CT:=CT-1

RG2(1)

CT=0

1

1

*Рисунок 1.2- Змістовний мікроалгоритм.*

**1.1.4 Таблиця станів регістрів.**

*Таблиця 1.1-Таблиця станів регістрів.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1🡪 | RG2🡪 | RG3 | СТ |
| П.С. | 0000000000000000 | 11111111100111**0** | 0111111111001101 | 1111 |
| 1🡪 | 0000000000000000 | 01111111110011**1** |  | 1110 |
| 2  🡪 | 0111111111001101  0011111111100110 | 10111111111001**1** |  | 1101 |
| 3  🡪 | + 0111111111001101  1011111110110011  0101111111011001 | 11011111111100**1** |  | 1100 |
| 4  🡪 | + 0111111111001101  1101111110100110  0110111111010011 | 01101111111110**0** |  | 1011 |
| 5🡪 | 0011011111101001 | 10110111111111**0** |  | 1010 |
| 6🡪 | 0001101111110100 | 11011011111111**1** |  | 1001 |
| 7  🡪 | + 0111111111001101  1001101111000001  0100110111100000 | 11101101111111**1** |  | 1000 |
| 8  🡪 | + 0111111111001101  1100110110101101  0110011011010110 | 11110110111111**1** |  | 0111 |
| 9  🡪 | + 0111111111001101  1110011010100011  0111001101010001 | 11111011011111**1** |  | 0110 |
| 10  🡪 | + 0111111111001101  1111001100011110  0111100110001111 | 01111101101111**1** |  | 0101 |
| 11  🡪 | + 0111111111001101  1111100101011100  0111110010101110 | 00111110110111**1** |  | 0100 |
| 12  🡪 | + 0111111111001101  1111110001111011  0111111000111101 | 10011111011011**1** |  | 0011 |
| 13  🡪 | + 0111111111001101  1111111000001010  0111111100000101 | 01001111101101**1** |  | 0010 |
| 14  🡪 | + 0111111111001101  1111111011010010  0111111101101001 | 00100111110110**1** |  | 0001 |
| 15  🡪 | + 0111111111001101  1111111100110110  0**111111110011011** | **000100111110110** |  | 0000 |

Округлення результату для його запису в задану розрядну мантису:

111111110011011000100111110110

+1

***111111110011011***100100111110110

*Mz=1, 111111110011011*

**1.1.5 Функціональна схема з відображенням управляючих сигналів.**

***x1***

***DR2***

***R***

***W1***

***SR1***

***DR=”0”***

***RG2***

***0***

***n-1***

***RG1***

***0***

***n***

***W2***

***SR2***

***n+1***

***n***

***x2***

***X***

***n***

***n***

***0***

***0***

***SM***

***n+1***

***n+1***

***CT***

***s***

***WCT***

***dec***

***W3***

***RG3***

***0***

***n***

***n***

***Y***

*Рисунок 1.3- Функціональна схема.*

**1.1.6 Закодований мікроалгоритм.**

*Таблиця 1.2-Таблиця кодування операцій і логічних умов.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| RG1:=0  RG2:=X  RG3:=Y  CT:=15  RG1:=RG1+RG3  RG1:=0.r(RG1)  RG2(n):=RG1[1].r(RG2)  CT:=CT-1 | R1  W2  W3  WCT  W1  SR1  SR2  dec |  | RG2[1]  CT=0 | X1  X2 |
|
|
| Початок |

R, W2, W3, WCT

Z1

Z2

1

X1

Z3

W1

Z4

X2

SR1,SR2,dec

1

Кінець

Z1

*Рисунок 1.4-Закодований мікроалгоритм.*

**1.1.7 Граф управляючого автомата Мура з кодами вершин.**

01

00

10

11

-

-

Z2

R W2, W3, WCT

Z1

0

Z4

SR1, SR3, dec

Z3

W1

\_\_

X1

\_\_

X2 X1

X1

\_\_ \_\_

X2 X1

X2

*Рисунок 1.5-Граф автомата Мура*

**1.1.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси не потрібна.

Знак мантиси: .

**1.1.9 Форма запису нормалізованого результату з плаваючою комою в пам’ять**

Pz=0.0001101 Mz=1, 111111110011011

Зн.Р P Зн.М M

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

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

**1.2 Другий спосіб множення**

**1.2.1 Теоретичне обгрунтування**

Перед початком множення другим способом множник Х записуються в регістр RG2, а множене Y – в молодші розряди регістру RG3 (тобто в регістрі установлюють Y0=Y2-n). В кожному і-му циклі множення додавання кодів RG3 і RG1 управляє цифра RG2(n), а в регістрі RG3 здійснюється зсув вліво на один розряд, в результаті чого формується величина Yі=2Yi-1. Оскільки сума часткових добутків в процесі множення нерухома, зсув в регістрі RG3 можна сполучити в часі з підсумуванням(як правило, tп ≥tз). В цьому випадку tм = ntп. Завершення операції множення визначається за нульовим вмістом регістру RG2, що також приводить до збільшення швидкодії, якщо множник ненормалізований.

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.

Вираз

подамо у вигляді

.

Отже, сума часткових добутків в i-му циклі, де , зводиться до обчислення виразу:

.

з початковими умовами *Z0=0, Y0=Y2-n, i=1.*

**1.2.2 Операційна схема**

***RG2(1)***

***2n***

***2n***

***1***

***1***

***SM***

***2n***

***RG2***

***RG1***

***1***

***2n***

***n***

***1***

***2n***

***RG3***

***1***

***2n***

*Рисунок 1.6-Операційна схема*

**1.2.3 Змістовний мікроалгоритм**

Початок

RG1:=0

RG2:=X

RG3:=Y

RG2(1)

1

RG1:=RG1+RG3

RG2:=0.r(RG2)

RG3:=l(RG3).0

RG2=0

1

Кінець

*Рисунок 1.7-Змістовний мікроалгоритм*

**1.2.4 Таблиця станів регістрів**

*Таблиця 1.3- Таблиця станів регістрів*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1 | RG2 🡪 | RG3 🡨 | C |
| П.С. | 000000000000000000000000000000 | 11111111100111**0** | 000000000000000111111111001101 | 0 |
| 1 | 000000000000000000000000000000 | 01111111110011**1** | 000000000000001111111110011010 | 0 |
| 2 | 000000000000001111111110011010 | 00111111111001**1** | 000000000000011111111100110100 | 0 |
| 3 | + 000000000000011111111100110100  000000000000101111111011001110 | 00011111111100**1** | 000000000000111111111001101000 | 0 |
| 4 | + 000000000000111111111001101000  000000000001101111110100110110 | 00001111111110**0** | 000000000001111111110011010000 | 0 |
| 5 | 000000000001101111110100110110 | 00000111111111**0** | 000000000011111111100110100000 | 0 |
| 6 | 000000000001101111110100110110 | 00000011111111**1** | 000000000111111111001101000000 | 0 |
| 7 | + 000000000111111111001101000000  000000001001101111000001110110 | 00000001111111**1** | 000000001111111110011010000000 | 0 |
| 8 | + 000000001111111110011010000000  000000011001101101011011110110 | 00000000111111**1** | 000000011111111100110100000000 | 0 |
| 9 | + 000000011111111100110100000000  000000111001101010001111110110 | 00000000011111**1** | 000000111111111001101000000000 | 0 |
| 10 | + 000000111111111001101000000000  000001111001100011110111110110 | 00000000001111**1** | 000001111111110011010000000000 | 0 |
| 11 | + 000001111111110011010000000000  000011111001010111000111110110 | 00000000000111**1** | 000011111111100110100000000000 | 0 |
| 12 | + 000011111111100110100000000000  000111111000111101100111110110 | 00000000000011**1** | 000111111111001101000000000000 | 0 |
| 13 | + 000111111111001101000000000000  001111111000001010100111110110 | 00000000000001**1** | 001111111110011010000000000000 | 0 |
| 14 | + 001111111110011010000000000000  011111110110100100100111110110 | 00000000000000**1** | 011111111100110100000000000000 | 0 |
| 15 | + 011111111100110100000000000000  111111110011011000100111110110 | **000000000000000** | 111111111001101000000000000000 | **1** |

Округлення результату для його запису в задану розрядну мантису:

111111110011011000100111110110

+1

***111111110011011***100100111110110

*Mz=1, 111111110011011*

**1.2.5 Функціональна схема з відображенням управляючих сигналів**

1

***x2***

***x1***

***W2***

***SR***

***DR=’0’***

***2n***

***1***

RG1

RG2

***1***

***n***

***R***

***W1***

***n***

***2n***

***X***

***2n***

***2n***

***1***

***1***

SM

***2n***

***2n***

***W3***

***SL***

***DL=’0’***

RG3

***1***

***2n***

***n***

***n***

***0***

***Y***

*Рисунок 1.8- Функціональна схема*

**1.2.6 Закодований мікроалгоритм**

*Таблиця 1.4 – Таблиця кодування мікрооперацій.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця 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  SR  SL |  | RG2[1]  RG2=0 | X1  X2 |
|  |

Початок

Кінець

R,W2,W3

W1

SR, SL

X1

X2

1

1

Z1

Z2

Z3

Z4

Z1

*Рисунок 1.9- Закодований мікроалгоритм.*

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

01

00

Z2

R W2, W3

Z1

0

Z4

SR, SL

Z3

W1

\_\_

X1

\_\_

X2 X1

X1

\_\_ \_\_

X2 X1

X2

-

-

10

11

*Рисунок 1.10- Граф автомата Мура*

**1.2.8 Обробка порядків і нормалізація**

Нормалізація мантиси не потрібна.

Знак мантиси:

**1.2.9 Форма запису нормалізованого результату з плаваючою комою в пам’ять**

Pz=0.0001101 Mz=1, 111111110011011

Зн.Р P Зн.М M

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

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

**1.3 Третій спосіб множення**

**1.3.1 Теоретичне обгрунтування**

Під час множення третім способом вага молодшого розряду RG3 дорівнює 2-2n, тому код у регістрі RG3 являє собою значення Y2-n. На початку кожного циклу множення здійснюється лівий зсув у регістрах RG1 і RG2, а потім виконується додавання, яким керує RG2(1). У результаті підсумування вмісту RG3 і RG1 може виникнути перенос у молодший розряд регістру RG2. У страшій частині суматора, на якому здійснюється підсумування коду RG2 з нулями, відбувається поширення переносу. Збільшення довжини RG2 на один розряд усуває можливість поширення переносу в розряди множника. Після виконання n циклів молодші розріди добутку будуть знаходитися в регістрі RG1, а старші – в регістрі RG2. Час множення третім способом визначається аналогічно першому способу і дорівнює ti=n(tї+tс)

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.

Вираз подамо у вигляді .

Отже, сума часткових добутків в *i*-му циклі, де , зводиться до обчислення виразу: .

з початковими умовами *Z0=0, i=1.*

**1.3.2 Операційна схема**

***n***

***2n+1***

***n+1***

***n+1***

***RG2***

***1***

***n+1***

***RG1***

***n***

***1***

***RG2(n)***

***RG3***

***CT***

***1***

***s***

***CT=0***

***2n+1***

***2n+1***

***1***

***1***

***SM***

***n***

***n***

***1***

**0**

*Рисунок 1.11-Операційна схема*

**1.3.3 Змістовний мікроалгоритм**

RG1:=0; RG2:=X;

RG3:=Y; CT:=15

Початок

RG2(n)

RG2.RG1:= RG2.RG1+RG3

1

RG2. RG1:=l(RG2. RG1).0

СT:=CT-1

ні

CT=0

так

Кінець

*Рисунок 1.12-Змістовний мікроалгоритм*

**1.3.4 Таблиця станів регістрів**

*Таблиця 1.5- Таблиця станів регістрів*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG2 | RG1 | RG3 | СТ |
| П.С. | 1111111110011100 | 000000000000000 | 111111111001101 | 1111 |
| 1  🡨 | 1111111100111001 | + 111111111001101  111111111001101  111111110011010 |  | 1110 |
| 2  🡨 | 1111111100111010  1111111001110101 | + 111111111001101  111111101100111  111111011001110 |  | 1101 |
| 3  🡨 | 1111111001110110  1111110011101101 | + 111111111001101  111111010011011  111110100110110 |  | 1100 |
| 4  🡨 | 1111110011101110  1111100111011101 | + 111111111001101  111110100000011  111101000000110 |  | 1011 |
| 5  🡨 | 1111100111011110  1111001110111101 | + 111111111001101  111100111010011  111001110100110 |  | 1010 |
| 6  🡨 | 1111001110111110  1110011101111101 | + 111111111001101  111001101110011  110011011100110 |  | 1001 |
| 7  🡨 | 1110011101111110  1100111011111101 | + 111111111001101  110011010110011  100110101100110 |  | 1000 |
| 8  🡨 | 1100111011111110  1001110111111101 | + 111111111001101  100110100110011  001101001100110 |  | 0111 |
| 9  🡨 | 1001110111111110  0011101111111100 | + 111111111001101  001101000110011  011010001100110 |  | 0110 |
| 10🡨 | 0111011111111000 | 110100011001100 |  | 0101 |
| 11🡨 | 1110111111110001 | 101000110011000 |  | 0100 |
| 12 | 1110111111110010  1101111111100101 | + 111111111001101  101000101100101  010001011001010 |  | 0011 |
| 13  🡨 | 1101111111100101  1011111111001100 | + 111111111001101  010001010010111  100010100101110 |  | 0010 |
| 14  🡨 | 1011111111001101  0111111110011011 | + 111111111001101  100010011111011  000100111110110 |  | 0001 |
| 15 | **1111111100110110** | **00100111110110**0 |  | **0000** |

Округлення результату для його запису в задану розрядну мантису:

111111110011011000100111110110

+1

***111111110011011***100100111110110

*Mz=1, 111111110011011*

**1.3.5 Функціональна схема з відображенням управляючих сигналів**

***n***

**0**

***1***

***2n+1***

***2n***

***2n***

***0***

***0***

***SM***

***2n+1***

***n+1***

***RG2***

***1***

***n***

***RG1***

***2n***

***0***

***RG3***

***n***

***R***

***W1***

***SL1***

***DL=’0’***

***W2***

***SL2***

***DL=’0’***

***W3***

***n***

***X***

***n***

***Y***

***CT***

***s***

1

***WCT***

***dec***

***x2***

*Рисунок 1.13-Функціональна схема*

**1.3.6 Закодований мікроалгоритм**

*Таблиця 1.6- Таблиця кодування мікрооперацій*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця 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  SL1  SL2  dec |  | RG2(n)  CT=0 | X1  X2 |
| Початок  Кінець  R,W2,W3,WCT  W1  SL1,SL2,dec  X1  X2  1  1  Z1  Z2  Z3  Z4  Z1 |

*Рисунок*

*1.14- Закодований мікроалгоритм*

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

11

01

00

10

Z2

R W2, W3

Z1

0

Z4

SR, SL

Z3

W1

\_\_

X1

\_\_

X2 X1

X1

\_\_ \_\_

X2 X1

X2

-

-

*Рисунок 1.15- Граф автомата Мура*

**1.3.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси не потрібна.

Знак мантиси: .

**1.3.9 Форма запису нормалізованого результату з плаваючою комою в пам’ять**

Pz=0.0001101 Mz=1, 111111110011011

Зн.Р P Зн.М M

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

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

**1.4 Четвертий спосіб множення**

**1.4.1 Теоретичне обґрунтування**

Перед множенням четвертим способом множник записують у регістр RG2, а множене – в старші розряди регістру RG3 (тобто в RG3 установлюють Y0=Y2-1). У кожному циклі цифра RG2(n), що знаходиться в старшому розряді регістру RG2, керує підсумуванням, а в RG3 здійснюється правий зсув на один розряд, що еквівалентно множенню вмісту цього регістра на 2-1. Час виконання множення четвертим способом складає ti=ntї, визначається аналогічно другому способу.

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.

Вираз подамо у вигляді .

Отже, сума часткових добутків в i-му циклі, де , зводиться до обчислення виразу: . з початковими значеннями i=1, Y0=2-1Y, Z0=0.

**1.4.2 Операційна схема**

***RG1***

***1***

***2n***

***2n***

***2n***

***1***

***1***

***SM***

***2n***

***2n***

***RG2(n)***

***RG2***

***1***

***n***

***2n***

***RG3***

***1***

***2n***

*Рисунок 1.16-Операційна схема*

**1.4.3 Змістовний мікроалгоритм**

Початок

RG1:=0

RG2:=X

RG3:=0.Y

RG2(n)

RG3:=0.r(RG3)

RG2:=l(RG2).0

Кінець

1

RG1:=RG1+RG3

1

RG2=0

*Рисунок 1.17-Змістовний мікроалгоритм*

**1.4.4 Таблиця станів регістрів**

*Таблиця 1.7- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG3 | RG2 |
| П.С. | 000000000000000000000000000000 | 011111111100110100000000000000 | 111111111001110 |
| 1 | 011111111100110100000000000000 | 001111111110011010000000000000 | 111111111001110 |
| 2 | + 001111111110011010000000000000  101111111011001110000000000000 | 000111111111001101000000000000 | 111111110011100 |
| 3 | + 001111111110011010000000000000  101111111011001110000000000000 | 000011111111100110100000000000 | 111111100111000 |
| 4 | + 000111111111001101000000000000  110111111010011011000000000000 | 000001111111110011010000000000 | 111111001110000 |
| 5 | + 000011111111100110100000000000  111011111010000001100000000000 | 000000111111111001101000000000 | 111110011100000 |
| 6 | + 000001111111110011010000000000  111101111001110100110000000000 | 000000011111111100110100000000 | 111100111000000 |
| 7 | + 000000111111111001101000000000  111110111001101110011000000000 | 000000001111111110011010000000 | 111001110000000 |
| 8 | + 000000011111111100110100000000  111111011001101011001100000000 | 000000000111111111001101000000 | 110011100000000 |
| 9 | + 000000001111111110011010000000  111111101001101001100110000000 | 000000000011111111100110100000 | 100111000000000 |
| 10 | + 000000000111111111001101000000  111111110001101000110011000000 | 000000000001111111110011010000 | 001110000000000 |
| 11 | 111111110001101000110011000000 | 000000000000111111111001101000 | 01110000000000 |
| 12 | 111111110001101000110011000000 | 000000000000011111111100110100 | 11100000000000 |
| 13 | + 000000000000111111111001101000  111111110010101000101100101000 | 000000000000001111111110011010 | 11000000000000 |
| 14 | + 000000000000011111111100110100  111111110011001000101001011100 | 000000000000000111111111001101 | 10000000000000 |
| 15 | + 000000000000001111111110011010  **111111110011011000100111110110** | 000000000000000011111111100110 | 00000000000000 |

Округлення результату для його запису в задану розрядну мантису:

111111110011011000100111110110

+1

***111111110011011***100100111110110

*Mz=1, 111111110011011*

**1.4.5 Функціональна схема з відображенням управляючих сигналів**

***RG1***

***1***

***2n***

***2n***

***2n***

***1***

***1***

***SM***

***2n***

***2n***

***RG2(n)***

***RG2***

***1***

***n***

***2n***

***RG3***

***1***

***2n***

***R***

***W3***

***SL***

***W1***

***W2***

***SR***

*Рисунок 1.18-Функціональна схема*

**1.4.6 Закодований мікроалгоритм**

*Таблиця 1.8- Таблиця кодування мікрооперацій*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця 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  SR  SL |  | RG2(n)  RG2=0 | X1  X2 |
|  |

Початок

R, W2, W3, ShR

W1

ShR,ShL

X1

X2

1

Z1

Z2

Z3

Z4

Кінець

Z1

1

*Рисунок 1.19- Закодований мікроалгоритм*

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

01

00

Z1

-

Q1Q2

01

00

-

X1

-

Z1

-

Z3

W1

Z4

ShR,ShL

Z2

Y1,Y2,Y3

Q3Q2Q1

-

Z2

R, W2, W3, ShR

X2

10

11

*Рисунок 1.20- Граф автомата Мура*

**1.4.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси не потрібна.

Знак мантиси: .

**1.4.9 Форма запису нормалізованого результату з плаваючою комою в пам’ять**

Pz=0.0001101 Mz=1, 111111110011011

Зн.Р P Зн.М M

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

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

**2. Операція ділення чисел**

Z=X: Y, X-ділене, Y – дільник.

**2.1 Перший спосіб ділення**

**2.1.1 Теоретичне обґрунтування**

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 RG2 (у вихідному стані в цьому регістрі записаний Х). Виходи RG2 підключені до входів СМ безпосередньо, тобто ланцюги видачі коду з RG2 не потрібні. Час для підключення n+1 цифри частки визначається виразом t=(n+1)(tд+tз), де tд - тривалість виконання мікрооперації додавання-віднімання; tз - тривалість виконання мікрооперації зсуву.

**2.1.2 Операційна схема**

М

***n+2***

***1***

***n+2***

***n+2***

***RG1***

***n+1***

***1***

***RG2***

***1***

***n***

CI

***SM***

***1***

***n+2***

***n+2***

***RG3***

***1***

***n+2***

*Рисунок 2.1-Операційна схема*

**2.1.3 Змістовний мікроалгоритм**

Початок

RG1:=0

RG2:=X;

RG3:=Y;

RG1:=l(RG1).)

RG2:=l(RG2).0

1

RG2(n+2)

RG2:=RG2++CI

RG2:=RG2+RG3

RG1(n+2)

1

Кінець

*Рисунок 2.2-Змістовний мікроалгоритм*

**2.1.4 Таблиця станів регістрів**

Так як Mx<My не виконується, тому перед початком ділення мантису діленого зсуваємо вправо, чим забезпечуємо зменшення її у два рази. Відповідно до порядку додається одиниця.

Px=8(10)+1(10)=9(10)=0.0001001(2) Mx = 1.011111111100111

*Таблиця 2.1- Таблиця станів регістрів*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 | МО |
| П.С. | 0000000000000000 | 00011111111100111 | 00111111111001101 |  |
| 1 | 000000000000000**1\*** | 00111111111001110  + 11000000000110011  00000000000000001 |  | 🡨 RG2🡨RG1  RG2:=RG2-RG3 |
| 2 | 00000000000000**1\***1 | 00000000000000010  + 11000000000110011  11000000000110101 |  | 🡨 RG2🡨RG1  RG2:=RG2-RG3 |
| 3 | 0000000000000**1\***10 | 10000000001101010  + 00111111111001101  11000000000110111 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 4 | 000000000000**1\***100 | 10000000001101110  + 00111111111001101  11000000000111011 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 5 | 00000000000**1\***1000 | 10000000001110110  + 00111111111001101  11000000001000011 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 6 | 0000000000**1\***10000 | 10000000010000110  + 00111111111001101  11000000001010011 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 7 | 000000000**1\***100000 | 10000000010100110  + 00111111111001101  11000000001110011 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 8 | 00000000**1\***1000000 | 10000000011100110  + 00111111111001101  11000000010110011 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 9 | 0000000**1\***10000000 | 10000000101100110  + 00111111111001101  11000000100110011 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 10 | 000000**1\***100000000 | 10000001001100110  + 00111111111001101  11000001000110011 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 11 | 00000**1\***1000000000 | 10000010001100110  + 00111111111001101  11000010000110011 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 12 | 0000**1\***10000000000 | 10000100001100110  + 00111111111001101  11000100000110011 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 13 | 000**1\***100000000000 | 10001000001100110  + 00111111111001101  11001000000110011 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 14 | 00**1\***1000000000000 | 10010000001100110  + 00111111111001101  11010000000110011 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 15 | 0**1\***10000000000000 | 10100000001100110  + 00111111111001101  11100000000110011 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 16 | **1\***100000000000000 | 11000000001100110  + 00111111111001101  00000000000110011 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |

**2.1.5 Функціональна схема з відображенням управляючих сигналів**

***W1***

***SL1***

***R***

***RG1***

***n+1***

***1***

***n+1***

***RG2***

**MX**

***n+2***

***y***

***β***

***DR***

***SM***

***0***

***RG3***

***n+1***

***W3***

**=1**

**n+2**

***n+1***

***n+1***

***0***

***0***

*α*

x

***RZ***

***1***

***n+1***

***0***

***W2***

***W4***

***SL2***

***W5***

***DL=’0’***

*Рисунок 2.3-Функціональна схема*

**2.1.6 Закодований мікроалгоритм**

*Таблиця 2.2- Таблиця кодування мікрооперацій*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | | |
| МО | УС |  | ЛУ | | Позначення |
| RG1:=0  RG2:=X;  RG3:=Y;  RG3:=l(RG3).  RG2:=l(RG2).0  RG2:=RG2++1  RG2:=RG2+RG1 | W1  W2  W3  SL1  SL2  W4,α,β  W5,α,β |  | RG2(n+1)  RG3(n+1) | | X1  X2 |
| Z1  Початок | |

W3, W2, W1

Z2

Z3

SL1, SL2

1

X1

W5,α,β

W4,α,β

Z4

Z5

X2

Z1

Кінець

1

*Рисунок 2.4- Закодований мікроалгоритм*

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

001

000

Z1

0

Z5

W5,α,β

Z4

W4,α,β

Z3

SL1, SL2

Z2

W3, W2,W1

-

*X2*

*X2*

*X1*

-

110

011

010

*Рисунок 2.5- Граф автомата Мура*

**2.1.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси не потрібна.

Знак мантиси: .

MZ=1,100000000000000

**2.1.9 Форма запису нормалізованого результату з плаваючою комою в пам’ять**

Pz=0.0000100 Mz=1,100000000000000

Зн.Р P Зн.М M

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

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

**2.2 Другий спосіб ділення(без відновлення остачі)**

**2.2.1 Теоретичне обґрунтування**

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.

**2.2.2 Операційна схема**

***2n***

***0***

***SM***

***RG3***

***RG2***

***RG1***

***p***

M

***0***

***n***

***2n***

***0***

***2n***

***0***

***0***

***2n***

*Рисунок 2.6-Операційна схема*

**2.2.3 Змістовний мікроалгоритм**

Початок

Кінець

RG2:=RG2+RG3

RG3:=0.r(RG3)

RG1:=l(RG1).SM(p)

RG2:=RG2++1

RG3:=0.r(RG3)

RG1:=l(RG1).SM(p)

RG2[2n+1]

RG1[n+1]

1

1

RG1:=0..01;

RG3:=Y;

RG2:=X

*Рисунок 2.7-Змістовний мікроалгоритм*

**2.2.4 Таблиця станів регістрів**

Так як Mx<My не виконується, тому перед початком ділення мантису діленого зсуваємо вправо, чим забезпечуємо зменшення її у два рази. Відповідно до порядку додається одиниця.

Px=8(10)+1(10)=9(10)=0.0001001(2) Mx = 1.011111111100111

*Таблиця 2.3- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG3 | RG2 | RG1 |
| П.С. | 000000000000000**1** | 001111111110011100000000000000 | 001111111110011010000000000000 |
| 1 | 0000000000000011 | 001111111110011100000000000000  + 110000000001100110000000000000  000000000000000010000000000000 | 000111111111001101000000000000 |
| 2 | 0000000000000110 | 000000000000000010000000000000  + 111000000000110011000000000000  111000000000110101000000000000 | 000011111111100110100000000000 |
| 3 | 0000000000001100 | 111000000000110101000000000000  + 000011111111100110100000000000  111100000000011011100000000000 | 000001111111110011010000000000 |
| 4 | 0000000000011000 | 111100000000011011100000000000  + 000001111111110011010000000000  111110000000001110110000000000 | 000000111111111001101000000000 |
| 5 | 0000000000110000 | 111110000000001110110000000000  + 000000111111111001101000000000  111111000000001000011000000000 | 000000011111111100110100000000 |
| 6 | 0000000001100000 | 111111000000001000011000000000  + 000000011111111100110100000000  111111100000000101001100000000 | 000000001111111110011010000000 |
| 7 | 0000000011000000 | 111111100000000101001100000000  + 000000001111111110011010000000  111111110000000011100110000000 | 000000000111111111001101000000 |
| 8 | 0000000110000000 | 111111110000000011100110000000  + 000000000111111111001101000000  111111111000000010110011000000 | 000000000011111111100110100000 |
| 9 | 0000001100000000 | 111111111000000010110011000000  + 000000000011111111100110100000  111111111100000010011001100000 | 000000000001111111110011010000 |
| 10 | 0000011000000000 | 111111111100000010011001100000  + 000000000001111111110011010000  111111111110000010001100110000 | 000000000000111111111001101000 |
| 11 | 0000110000000000 | 111111111110000010001100110000  + 000000000000111111111001101000  111111111111000010000110011000 | 000000000000011111111100110100 |
| 12 | 0001100000000000 | 111111111111000010000110011000  + 000000000000011111111100110100  111111111111100010000011001100 | 000000000000001111111110011010 |
| 13 | 0011000000000000 | 111111111111100010000011001100  + 000000000000001111111110011010  111111111111110010000001100110 | 000000000000000111111111001101 |
| 14 | 0110000000000000 | 111111111111110010000001100110  + 000000000000000111111111001101  111111111111111010000000110011 | 000000000000000011111111100110 |
| 15 | 1100000000000000 | 111111111111111010000000110011  + 000000000000000011111111100110  111111111111111110000000011001 | 000000000000000001111111110011 |

**2.2.5 Функціональна схема з відображенням управляючих сигналів**

***0***

***n***

***RG1***

***R***

***SL***

***W2***

***W3***

***W4***

***2n***

***0***

***RG2***

***2n***

***0***

***0***

***2n***

***SM***

***DL=p***

**MX**

2n+1

2n+1

β

***2n***

***0***

***RG3***

***W1***

***SR***

***DR=’0’***

2n+1

=1

y

x

α

*Рисунок 2.8-Функціональна схема*

**2.2.6 Закодований мікроалгоритм**

*Таблиця 2.4- Таблиця кодування мікрооперацій*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| RG1:=0..01;  RG3:=Y;  RG2:=X  RG2:=RG2+RG1  RG1:=0.r(RG1)  RG3:=l(RG3).SM(p)  RG2:=RG2++1 | R  W1  W2  W3,α,β  SR  SL  W4,α,β |  | RG2[2n+1]  RG1[n+1] | X1  X2 |
| 1  1  Z3  Початок  Кінець  W3,SR,SL,α,β  W4,SR,SL,α,β  X1  X2  R,W1,W2  Z1  Z2  Z4  Z1 |

*Рисунок 2.9- Закодований мікроалгоритм*

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

-

Z1

0

Z2

R, W1, W2

Z4

W4, SR, SL,α,β

Z3

W3, SR, SL,α,β

*X2*

*X1*

*X2*

00

01

11

10

*Рисунок 2.10- Граф автомата Мура*

**2.2.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси не потрібна.

Знак мантиси: .

MZ=1,100000000000000

**2.2.9 Форма запису нормалізованого результату з плаваючою комою в пам’ять**

Pz=0.0000100 Mz=1,100000000000000

Зн.Р P Зн.М M

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

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

**3. Операція додавання чисел**

**Z=X+Y.**

**3.1 Теоретичне обґрунтування способу**

В пам’яті числа зберігаються у ПК. На першому етапі додавання чисел з плаваючою комою виконують вирівнювання порядків до числа із старшим порядком. На другому етапі виконують додавання мантис. Додавання мантис виконується у доповнювальних кодах, при необхідності числа у ДК переводяться в АЛП. Додавання виконується порозрядно на n-розрядному суматорі з переносом. Останній етап – нормалізація результату. Виконується за допомогою зсуву мантиси результату і коригування порядку результату. Порушення нормалізації можливо вліво і вправо, на 1 розряд вліво і на n розрядів вправо.

**Виконання етапів вирівнювання порядків і додавання мантис:**

1. Порівняння порядків.

,

.

1. Вирівнювання порядків.

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

*Таблиця 3.1- Таблиця зсуву мантиси на етапі вирівнювання порядків*

|  |  |  |
| --- | --- | --- |
|  |  | Мікрооперація |
| 0.111111111001101 | 11 | П.С. |
| 0.011111111100110 | 10 | 🡪 |
| 0.001111111110011 | 01 | 🡪 |
| 0.000111111111001 | 00 | 🡪 |

1. Додавання мантис у модифікованому ДК.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1**  + | **1,** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **1** | **0** | **0** | **1** | **0** |
| **0** | **0,** | **0** | **0** | **0** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **0** | **0** | **1** |
| **1** | **1,** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **1** | **1** |

Рисунок 3.1-Додавання мантис

1. Перевірка на нульовий результат

нормалізація

0

1

Кінець

Початок

RG1:=Mz;

M

RG1

Mz

n

1

1

n-2

1

n-2

Рисунок 3.2-Операційна схема Рисунок 3.2-Операційна схема

|  |  |  |
| --- | --- | --- |
| RG1 | M | |
| 001000000101011 | 1 |

*Таблиця 3.2-* Перевірка на нульовий результат

1. Нормалізація результату (В ПК).

**3.2 Операційна схема**

m-кількість розрядів мантиси

n-кількість розрядів порядку

q=]log2m[

R

L

*КС*

CT=0

m+1 RGZ 0

q CT 1

n+1 RGPZ 0

Рисунок 3.3-Операційна схема

Виконаємо синтез КС для визначення порушення нормалізації.

*Таблиця 3.2-Визначення порушення нормалізації*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Розряди регістру 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 |

L= Z0, R=.

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

**3.3 Змістовний мікроалгоритм**

RGZ:=RGZ(m+2).r(RGZ)

RGPZ:=RGPZ+1

1

1

1

0

0

Кінець

0

RGZ:=l(RGZ).0

RGPZ:=RGPZ-1

CT:=CT-1;

Початок

CT:=m; RGZ:=Z;

*Рисунок 3.4-Змістовний мікроалгоритм*

**3.4 Таблиця станів регістрів**

*Таблиця 3.3- Таблиця станів регістрів*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ такту** | **RGPZ** | **RGZ** | **ЛПН(L)** | **ППН(R)** | **СT** | **Мікрооперація** |
| **ПС** | 001000 | 00,110111111010101 | 0 | 0 | 100 |  |

**3.5 Функціональна схема з відображенням управляючих сигналів**

R

L

Z0

ShL

ShR

W1,W2

m+1 m m-1 RGZ 0

q CT 1

CT=0

inc

dec

W

W

dec

n+1 RGPZ 0

*Рисунок 3.5Функціональна схема*

**3.6 Закодований мікроалгоритм**

*Таблиця 3.4- Таблиця кодування мікрооперацій*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| CT:=m;  RGZ:=Z;  Z’0 Z0:=  RGZ:=RGZ(m+2).r(RGZ)  RGPZ:=RGPZ+1  RGZ:=l(RGZ).0  RGPZ:=RGPZ-1  CT:=CT-1; | W  W1  W2  ShR  inc  ShL  dec  dec |  | Z’0 =0  0 | X1  X2  X3  X4 |
|  |

Z1

Початок

Z2

W,W1

X1

1

Z3

W2

1

ShR,inc

Z4

1

Z5

ShL,dec,dec

X4

1

Кінець

Z6

*Рисунок 3.6- Закодований мікроалгоритм*

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

Z1

-

Q3Q2Q1

000

-

001

Z2

W,W1

X4

110

Z5

ShL,dec,dec

-

Z3

W2

X2

011

Z4

ShR,inc

X2

010

*Рисунок 3.7- Граф автомата Мура*

**3.8 Форма запису нормалізованого результату з плаваючою комою в пам’ять**

Pz=0.0001000 Mz=1,11011111101010101

Зн.Р P Зн.М M

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

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

**4. Операція добування кореня**

**Z=**

**4.1 Теоретичне обґрунтування способу**

Алгоритм обчислення квадратного кореня з n-розрядної мантиси числа зводиться до підбору цифр результату розряд за розрядом, розпочинаючи зі старшого 2-1 розряду. За цього обчислення і-ї цифри результату Z відбувається наступним чином. Після отримання чергової (і-1)-ї цифри aі-1 в і-й розряд А розміщується одиниця. Аргумент вводиться зі старших розрядів. Порядок результату дорівнює поділеному на два порядку аргумента. З мантиси добувається корінь завдяки нерівностям:

; ;

.

Виконання операції зводиться до послідовності дій:

1. Одержання остачі. ;

2. Якщо , то .

3. Якщо , то .

Відновлення остачі додає зайвий такт, але можна зробити інакше: , тоді корінь добувається без відновлення залишку. Для цього зсувається на 2 розряди ліворуч, а - на 1 розряд ліворуч, і формується як при діленні.

**4.2 Операційна схема**

🡨

RG1

🡨

n+2

1

=1

RG2

RG3

🡨

n

1

n

1

SM

n+2

1

n+2

1

“1”

2

n+2

n

n+2

CT

s

1

“q”

2

2

*Рисунок 4.1-Операційна схема*

**4.3 Змістовний мікроалгоритм**

Початок

RG3:=X;

RG2:=0..0;

RG1:=0..0

CT:=15

Кінець

RG2:=RG2+RG1.11

RG2:=RG2+.11

RG2[n+2]

1

CT=0

1

RG1:=L(RG1).

CT:=CT-1

RG2:=L2[RG2].RG3(n).RG3(n-1)

RG3:=L2[RG3].0.0

*Рисунок 4.2-Змістовний мікроалгоритм*

**4.4 Таблиця станів регістрів**

*Таблиця 4.1- Таблиця станів регістрів*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 | CT |
| П.С. | 000000000000000 | 00000000000000000 | 111111111001110 | 1111 |
| 1🡨  🡨 | 000000000000001 | 00000000000000011  + 11111111111111111  00000000000000010 | 111111100111000 | 1110 |
| 2🡨  🡨 | 000000000000011 | 00000000000001011  + 11111111111111011  00000000000000110 | 111110011100000 | 1101 |
| 3🡨  🡨 | 000000000000111 | 00000000000011011  + 11111111111110011  00000000000001110 | 111001110000000 | 1100 |
| 4🡨  🡨 | 000000000001111 | 00000000000111011  + 11111111111100011  00000000000011110 | 100111000000000 | 1011 |
| 5🡨  🡨 | 000000000011111 | 00000000001111010  + 11111111111000011  00000000000111101 | 011100000000000 | 1010 |
| 6🡨  🡨 | 000000000111111 | 00000000011110101  + 11111111110000011  00000000001111000 | 110000000000000 | 1001 |
| 7🡨  🡨 | 000000001111111 | 00000000111100011  + 11111111100000011  00000000011100110 | 000000000000000 | 1000 |
| 8🡨  🡨 | 000000011111111 | 00000001110011000  + 11111111000000011  00000000110011011 | 000000000000000 | 0111 |
| 9🡨  🡨 | 000000111111111 | 00000011001101100  + 11111110000000011  00000001001101111 | 000000000000000 | 0110 |
| 10🡨  🡨 | 000001111111111 | 00000100110111100  + 11111100000000011  00000000110111111 | 000000000000000 | 0101 |
| 11🡨  🡨 | 000011111111110 | 00000011011111100  + 11111000000000011  11111011011111111 | 000000000000000 | 0100 |
| 12🡨  🡨 | 000111111111100 | 11101101111111100  + 00001111111111011  11111101111110111 | 000000000000000 | 0011 |
| 13🡨  🡨 | 001111111111001 | 11110111111011100  + 00011111111110011  00010111111001111 | 000000000000000 | 0010 |
| 14🡨  🡨 | 011111111110011 | 01011111100111100  + 11000000000011011  00011111101010111 | 000000000000000 | 0001 |
| 15🡨  🡨 | 111111111100110 | 01111110101011100  + 10000000000110011  11111110110001111 | 000000000000000 | 0000 |

**4.5 Функціональна схема з відображенням управляючих сигналів**

SM

n+2

1

n+2

1

“1”

CT

WCT

dec

1

s

2

=1

n

n+2

α

“q”

R2

SLL2

W2

W3

🡨

n+2

1

RG2

x

W1

R

SLL3

RG3

1

n

🡨

2

2

RG1

R1

SL1

🡨

n

1

&

n+2

*Рисунок 4.3-Функціональна схема*

**4.6 Закодований мікроалгоритм**

*Таблиця 4.2- Таблиця кодування мікрооперацій*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| RG1:=0..0;  RG2:=0..0;  RG3:=X  CT:=15  RG2:=RG2+RG1.11  RG2:=RG2+.11  RG2=LL(RG2).RG3(n).RG2(n-1)  RG3:=LL(RG3).00  RG1:=L(RG1).  CT:=CT-1 | R1  R2  W1  WCT  W2,α  W3,α  SLL2  SLL3  SL1  dec |  | RG2[n+2]  CT=0 | X1  X2 |
| SL1  dec  Кінець  X2  1  W2,α  W3,α  SLL2, SLL3  Початок  W1, R2, R1,WCT  X1  1  Z1  Z4  Z5  Z3  Z2  Z6  Z1 |

*Рисунок 4.4- Закодований мікроалгоритм*

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

Z5

W3,α

Z6

SL1

000

001

011

010

110

100

Z1

0

-

X2

-

-

X1

-

Z2

W1,R2, R1, WCT

Z3

SLL2, SLL3

Z4

W2,α

*Рисунок 4.5- Граф автомата Мура*

**4.8 Обробка порядків**

PZ=PX/2=8/2=410=1002.

**4.9 Форма запису нормалізованого результату з плаваючою комою в пам’ять**

Pz=0.0000100 Mz=0,111111111100110

Зн.Р P Зн.М M

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

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

**5. Синтез управляючого автомату для операційного пристрою ділення другим способом.**

x3x2x1=110=6 – операція ділення другим способом.

**5.1 Таблиця співвідношення управляючих входів операційного автомата і виходів управляючого автомата**

За закодованим мікроалгоритмом (Рис. 1.14) складемо таблицю:

*Таблиця 5.1 Таблиця кодування сигналів*

|  |  |
| --- | --- |
| Входи операційного автомата | Виходи управляючого автомата |
| R, W1, W2 | Y1 |
| W3, SR, SL,α,β | Y2 |
| W4, SR, SL,α,β | Y3 |

**5.2 Мікроалгоритм в термінах управляючого автомата**

Зробимо автомат Мура циклічним задля зменшення кількості вершин.

1

1

Z3

Початок

Кінець

Y2

Y3

X1

X2

Y1

Z1

Z2

Z4

Z1

*Рисунок 5.1- Закодований мікроалгоритм*

Будуємо граф автомата Мура

11

10

-

Z1

0

Z2

Y1

Z4

Y3

Z3

Y2

*X2*

*X1*

*X2*

00

01

Q2Q1

*Рисунок 5.2- Граф автомата Мура*

**5.3 Структурна таблиця автомата**

За графом автомата мура складаємо структурну таблицю автомата. Значення функцій збудження тригерів визначаються відповідно до графічної схеми переходів T-тригера.

*Таблиця 5.2-Структурна таблиця автомата*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Перехід | Q2Q1 | Q2Q1 | x1x2 | y1y2y3 | T2 | T1 |
| *z1z2* | 00 | 01 | -- | 000 | 0 | 1 |
| *z2 z3* | 01 | 11 | 0- | 100 | 1 | 0 |
| *z2 z4* | 01 | 10 | 1- | 100 | 1 | 1 |
| *z3 z3* | 11 | 11 | 00 | 010 | 0 | 0 |
| *z3 z4* | 11 | 10 | 10 | 010 | 0 | 1 |
| *z3 z1* | 11 | 00 | -1 | 010 | 1 | 1 |
| *z4 z4* | 10 | 10 | 10 | 001 | 0 | 0 |
| *z4 z3* | 10 | 11 | 00 | 001 | 0 | 1 |
| *z4 z1* | 10 | 00 | -1 | 001 | 1 | 0 |

T-тригер:

**5.4 Синтез функцій виходів і переходів**

0

1

0

0

Q2

Q1

Y3

1

0

0

0

Q2

Q1

Y2

0

0

1

0

Q2

Q1

Y1

0

1

1

0

0

1

1

0

1

1

0

0

1

1

0

0

Q2

Q1

X2

X1

T2

0

1

0

1

1

1

-

0

0

1

1

1

1

0

0

1

1

Q2

Q1

X2

X1

T1

*Рисунок 5.3- Діаграми Вейча*

**5.5 Функціональна схема пристрою (виходи управляючого автомата**

**підключені до входів операційного автомата)**

­­

T2

R

T

C

S

G

“1”

T1

R

T

C

S

“1”

&

1

T2

&

&

&

Y1

Y2

Y3

&

&

&

&

T1

K1

1

***0***

***n***

***RG1***

***Y2***

***Y3***

***Y1***

***Y2***

***Y3***

***2n***

***0***

***RG2***

***2n***

***0***

***0***

***2n***

***SM***

***DL=p***

***MX***

***2n+1***

***2n+1***

***Y2β***

***2n***

***0***

***RG3***

***Y1***

***Y2***

***Y3***

***2n+1***

***=1***

***y***

***Y1***

***Y3***

*Рисунок 5.5- Функціональна схема пристрою*

**Висновки**

В данній розрахунковій роботі було переведено номер залікової книжки в двійкову систему числення і за допомогою цього записано два числа: X2 і Y2.

Числа  і  записано в прямому коді у формі з плаваючою комою (з порядком, характеристикою та мантисою), як вони зберігаються у пам’яті. На порядок і характеристику відведено 8 розрядів, на мантису 16 розрядів (з урахуванням знакових розрядів).

Виконано 8 операцій з числами  і  з плаваючою комою: чотири способи множення, два способи ділення, додавання та добування кореня з . Для обробки мантис кожної операції, подано: теоретичне обґрунтування способу, операційну схему, змістовний мікроалгоритм, таблицю станів регістрів (лічильника), довжина яких забезпечує одержання 15 основних розрядів мантиси результату, функціональну схему з відображенням управляючих сигналів, закодований мікроалгоритм, граф управляючого автомата Мура з кодами вершин, обробку порядків (показати у довільній формі), форму запису нормалізованого результату з плаваючою комою в пам’ять. Вказані пункти виконані для етапу нормалізації результату з урахуванням можливого нулевого результату.

Для операції з двійковим номером  побудувано управляючий автомат Мура на T-тригерах і елементах булевого базису.