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-44

Онищенко Артемій Олександрович  
  
Керівник: Клименко І.А.

2015 р.

**ЗМІСТ**

Обґрунтування варіанту………………………………………………………......6

1 .Операція множення чисел……………………………………………………...6

1.1 Перший спосіб множення…………………………………...…….…...6

1.1.1 Теоретичне обґрунтування способу…………………….…...6

1.1.2 Операційна схема…………………………………..................7

1.1.3 Змістовний мікроалгоритм………………………..................7

1.1.4 Таблиця станів регістрів…………………………..................8

1.1.5 Функціональна схема пристрою……………………….…….9

1.1.6 Закодований мікроалгоритм……………………...................9

1.1.7 Граф управляючого автомата………………….……………10

1.1.8 Обробка порядків ……………………………….…………..10

1.1.9 Форма подання результату в пам’ятті……………………...10

1.2 Другий спосіб множення………….………………………...………...11

1.2.1 Теоретичне обґрунтування способу………………………...11

1.2.2 Операційна схема………………………………….................11

1.2.3 Змістовний мікроалгоритм……………………….................12

1.2.4 Таблиця станів регістрів…………………………..................12

1.2.5 Функціональна схема пристрою…………………………….13

1.2.6 Закодований мікроалгоритм……………………...................13

1.2.7 Граф управляючого автомата………………….…………....14

1.2.8 Обробка порядків ……………………………….…………...15

1.2.9 Форма подання результату в пам’ятті……………………...15

1.3 Третій спосіб множення……………...……………………...………..16

1.3.1 Теоретичне обґрунтування способу………………….……..16

1.3.2 Операційна схема………………………………….................16

1.3.3 Змістовний мікроалгоритм……………………….................17

1.3.4 Таблиця станів регістрів…………………………...................17

1.3.5 Функціональна схема пристрою………………………….….18

1.3.6 Закодований мікроалгоритм……………………....................19

1.3.7 Граф управляючого автомата………………….…………….19

1.3.8 Обробка порядків ……………………………….……………19

1.3.9 Форма подання результату в пам’ятті……………………....20

1.4 Четвертий спосіб множення………………………………….......…...……...20

1.4.1 Теоретичне обґрунтування способу…………………….…...20

1.4.2 Операційна схема…………………………………..................20

1.4.3 Змістовний мікроалгоритм………………………...................21

1.4.4 Таблиця станів регістрів…………………………...................21

1.4.5 Функціональна схема пристрою………………………..…….22

1.4.6 Закодований мікроалгоритм……………………....................22

1.4.7 Граф управляючого автомата………………….…….……….23

1.4.8 Обробка порядків ……………………………….……………24

1.4.9 Форма подання результату в пам’ятті……………………....24

2 Операція ділення чисел………………………………………………………...24

2.1 Перший спосіб ділення……………………………………...………...24

2.1.1 Теоретичне обґрунтування способу………………………...24

2.1.2 Операційна схема………………………………….................24

2.1.3 Змістовний мікроалгоритм………………………..................25

2.1.4 Таблиця станів регістрів…………………………..................25

2.1.5 Функціональна схема пристрою……………………………..27

2.1.6 Закодований мікроалгоритм……………………...................27

2.1.7 Граф управляючого автомата………………….…………….28

2.1.8 Обробка порядків ……………………………….……………28

2.1.9 Форма подання результату в пам’ятті……………………....29

2.2 Другий спосіб ділення….…………………………………...……….....29

1.1.1 Теоретичне обґрунтування способу……………………….....29

1.1.2 Операційна схема…………………………………...................29

1.1.3 Змістовний мікроалгоритм………………………...................30

1.1.4 Таблиця станів регістрів…………………………...................30

1.1.5 Функціональна схема пристрою…………………………..…32

1.1.6 Закодований мікроалгоритм……………………....................32

1.1.7 Граф управляючого автомата………………….………….…33

1.1.8 Обробка порядків ……………………………….……………33

1.1.9 Форма подання результату в пам’ятті……………………....34

3 Операція додавання чисел……………………………….…………………...…34

3.1 Теоретичне обґрунтування способу………………………………......34

3.2 Операційна схема…………………………………...............................34

3.3 Змістовний мікроалгоритм………………………...............................35

3.4 Таблиця станів регістрів…………………………................................36

3.5 Функціональна схема пристрою………………………………….…...36

3.6 Закодований мікроалгоритм…………………….................................36

3.7 Граф управляючого автомата………………….…………………..….37

3.8 Обробка порядків ……………………………….………………..……37

3.9 Форма подання результату в пам’ятті……………………………......38

4 Операція добування кореня…………………………………………………….38

4.1 Теоретичне обґрунтування способу……………………………..…....38

4.2 Операційна схема…………………………………...............................39

4.3 Змістовний мікроалгоритм………………………...............................39

4.4 Таблиця станів регістрів…………………………...............................40

4.5 Функціональна схема пристрою………………………………….…..41

4.6 Закодований мікроалгоритм……………………................................41

4.7 Граф управляючого автомата………………….…………………….42

4.8 Обробка порядків ……………………………….……………………43

4.9 Форма подання результату в пам’ятті……………………………....43

5 Синтез управляючого автомата для операційного пристрою ……………...43

5.1 Таблиця співвідношення управляючих входів операційного автомата і виходів управляючого автомата …………………………………………..43

5.2 Мікроалгоритм в термінах управляючого автомата………….............43

5.3 Структурна таблиця автомата………………………………….............44

5.4 Синтех функцій виходів і переходів………………………….............. 44

5.5 Функціональна схема пристою ………………………………..............46

Висновок……………………………………………………………………………47

**Обґрунтування варіанту**

Номер залікової книжки: 441910=10001010000112

X2=- 10110101,0000111

Y2= +10110,1010000111

**Виконання роботи**

***Завдання 1***

В прямому коді:

В прямому коді:

X**2:** Зн.Р P=+00010002 Зн.М M= -,101101010000111

Y2: Зн.Р P=+00001012 Зн.М M= +,101101010000111

E=P+28

X2: Зн.Е Е=100010002 Зн.М M= -,101101010000111

Y2: Зн.Р Е=100001012 Зн.М M= +,101101010000111

***Завдання 2***

**1. Операція множення чисел**.

Z=YX, де Y-множене, X-множник.

**1.1 Перший спосіб множення.**

**1.1.1 Теоретичне обгрунтування.**

Числа множаться у прямих кодах. Під час множення чисел 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 двох чисел Y та X може бути реалізоване шляхом виконання визначеного циклічного процесу, характер якого залежить від конкретної форми заданого виразу.

Множення мантис першим способом здійснюється з молодших розрядів множника, сума часткових добутків зсувається вправо, а множене залишається нерухомим. Під час множення першим способом в першому такті і-го циклу аналізується значення RG2(n) — молодшого n-го розряду регістру RG2, в якому знаходиться чергова цифра множника. Вміст RG3 додається до суми часткових добутків, що зняходяться в регістрі RG1, якщо RG2(n)=1, або не додається, якщо RG2(n)=0. В другому такті здійснюється правий зсув у регістрах RG1 і RG2, що еквівалентно множенню їх вмісту на 2(-1). За зсуву 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 — в RG1. Час множення, якщо не застосовуються методи прискорення операції, визначається виразом:

*,*

де tП і tЗ – тривалості тактів підсумовування і зсуву відповідно.

Добуток двох чисел представляється у вигляді:

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

.

**1.1.2 Операційна схема.**

***RG2(n)***

***RG2***

***RG1***

***0***

***n***

***1***

***n***

***n***

***0***

***0***

***n***

***n***

***SM***

***n***

***n***

***n***

***RG3***

***0***

***CT***

***q***

***1***

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

CT:=CT-1

RG2(n)

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. | 00000000000000000 | 101101010000111 | 10110101000011 | 1111 |
| 1🡪 | 00010110101000011 | 110110101000011 |  | 1110 |
| 2🡪 | +  00101101010000111  1000011111001010  0100001111100101 | 011011010100001 |  | 1101 |
| 3🡪 | +  00101101010000111  1001111001101100  0100111100110110 | 001101101010000 |  | 1100 |
| 4🡪 | 0010011110011011 | 000110110101000 |  | 1011 |
| 5🡪 | 0001001111001101 | 100011011010100 |  | 1010 |
| 6🡪 | 0000100111100110 | 110001101101010 |  | 1001 |
| 7🡪 | 0000010011110011 | 011000110110101 |  | 1000 |
| 8🡪 | +  00101101010000111  0101111101111010  0010111110111101 | 001100011011010 |  | 0111 |
| 9🡪 | 0001011111011110 | 100110001101101 |  | 0110 |
| 10🡪 | +  00101101010000111  0111001001100101  0011100100110010 | 110011000110110 |  | 0101 |
| 11🡪 | 0001110010011001 | 011001100011011 |  | 0100 |
| 12🡪 | +  00101101010000111  0111011100100000  0011101110010000 | 001100110001101 |  | 0011 |
| 13🡪 | +  00101101010000111  1001011000010111  0100101100001011 | 100110011000110 |  | 0010 |
| 14🡪 | 0010010110000101 | 110011001100011 |  | 0001 |
| 15🡪 | **+**  00101101010000111  1000000000001100  0100000000000110 | 011001100110001 |  | 0000 |

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

***x1***

***n***

***R***

***W1***

***SR1***

***RG2***

***n***

***1***

***RG1***

***0***

***W2***

***SR2***

***n***

***n+1***

***x2***

***X***

***0***

***0***

***SM***

*=* ***1***

***n***

***n***

***n+1***

***n+1***

***CT***

***q***

***1***

***WCT***

***D***

***RG3***

***0***

***n***

***q***

***n***

***W3***

***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:=RG1(n).r(RG2)  CT:=CT-1 | R  W2  W3  WCT  W1  SR1  SR2  D |  | RG2(n)  CT=0 | X1  X2 |
|
|
| Початок |

Z1

Z2

R, W2, W3, WCT

1

X1

Z3

W1

Z4

X2

SR1,SR2,D

1

Кінець

Z5

*Рисунок 1.4-Закодований мікроалгоритм.*

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

Q3Q2Q1

001

000

-

Z1

-

Z2

R, W2, W3, WCT

110

X2

Z5

**-**

X1

Z4

SR1, SR2,D

-

Z3

W1

010

011

*Рисунок 1.5-Граф автомата Мура*

**1.1.8 Обробка порядків і нормалізація**

.   
Знак мантиси: .

Нормалізація мантиси не потрібна.

MZ= , 1000000000001100

.

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

Зн.Р P=+1310 Зн.М M

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

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

**1.2 Другий спосіб множення**

**1.2.1 Теоретичне обгрунтування**

Числа множаться у прямих кодах. Під час множення чисел 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 другим способом здійснюється з молодших розрядів, множене зсувається вліво, а сума часткових добутків залишається нерухомою. Перед початком множення другим способом множник X записують у регістр RG2, а множене Y – в молодші розряди регістру RG3(тобто в регістрі RG3 установлюють Y0=Y2-n ). В кожному i-му циклі множення додаванням кодів RG3 і RG1 керує цифра RG2(n), а в регістрі RG3 здійснюється зсув вліво на один розряд, у результаті чого формується величина Yi=2Yi-1 . Оскільки сума часткових добутків у процесі множення нерухома, зсув у регістрі RG3 можна сполучити в часі з підсумовуванням (як правило, tП>tЗ). Завершення операції множення визначається за нульовим вмістом регістру RG2, що також приводить до збільшення швидкодії, якщо множник ненормалізований.

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(n)***

***1***

***n***

***RG2***

***1***

***n***

***RG1***

***2n***

***1***

***2n***

***1***

***1***

***2n***

***2n***

***SM***

***2n***

***1***

***2n***

***RG3***

*Рисунок 1.6-Операційна схема*

**1.2.3 Змістовний мікроалгоритм**

Початок

RG1:=0

RG2:=X

RG3:=Y

RG2(n)

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🡨 |
| П.С. | 0000000000000000000000000000000 | 101101010000111 | 0000000000000000101101010000111 |
| 1 | 0000000000000000101101010000111 | 010110101000011 | 0000000000000001011010100001110 |
| 2 | +  0000000000000001011010100001110  000000000000010000111110010101 | 001011010100001 | 0000000000000010110101000011100 |
| 3 | +  0000000000000010110101000011100  000000000000100111100110110001 | 000101101010000 | 0000000000000101101010000111000 |
| 4 | 000000000000100111100110110001 | 000010110101000 | 0000000000001011010100001110000 |
| 5 | 000000000000100111100110110001 | 000001011010100 | 0000000000010110101000011100000 |
| 6 | 000000000000100111100110110001 | 000000101101010 | 0000000000101101010000111000000 |
| 7 | 000000000000100111100110110001 | 000000010110101 | 0000000001011010100001110000000 |
| 8 | +  0000000001011010100001110000000  000000001011111011110100110001 | 000000001011010 | 0000000010110101000011100000000 |
| 9 | 000000001011111011110100110001 | 000000000101101 | 0000000101101010000111000000000 |
| 10 | +  0000000101101010000111000000000  000000111001001100101100110001 | 000000000010110 | 0000001011010100001110000000000 |
| 11 | 000000111001001100101100110001 | 000000000001011 | 0000010110101000011100000000000 |
| 12 | +  0000010110101000011100000000000  000011101110010000001100110001 | 000000000000101 | 0000101101010000111000000000000 |
| 13 | +  0000101101010000111000000000000  001001011000010111001100110001 | 000000000000010 | 0001011010100001110000000000000 |
| 14 | 001001011000010111001100110001 | 000000000000001 | 0010110101000011100000000000000 |
| 15 | +  0010110101000011100000000000000  100000000000110011001100110001 | 000000000000000 | 0101101010000111000000000000000 |

**1.2.5 Функціональна схема з відображенням управляючих сигналів**

***x2***

***x1***

***=1***

***2n***

***1***

RG1

RG2

***n***

***1***

***R***

***W1***

***W2***

***SR***

***n***

***2n***

***X***

***1***

***1***

***2n***

***2n***

SM

***2n***

***2n***

RG3

***2n***

***1***

***W3***

***SL***

***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(RG2)  RG3:=l(RG3).0 | R  W2  W3  W1  SR  SL |  | RG2(n)  RG2=0 | X1  X2 |
|  |

Початок

Кінець

R,W2,W3

W1

SR, SL

X1

X2

1

1

Z1

Z2

Z3

Z4

Z5

*Рисунок 1.9- Закодований мікроалгоритм.*

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

Q3Q2Q1

Z1

-

01

00

-

X1

-

Z1

-

Z3

W1

Z4

SR,SL

Z2

Y1,Y2,Y3

001

000

-

Z2

R,W2 W3

110

Z5

**-**

X2

011

010

*Рисунок 1.10- Граф автомата Мура*

**1.2.8 Обробка порядків і нормалізація**

.   
Знак мантиси: .

Нормалізація мантиси не потрібна.

MZ= , 1000000000001100

.

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

Зн.Р P=+1310 Зн.М M

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

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

**1.3 Третій спосіб множення**

**1.3.1 Теоретичне обгрунтування**

Числа множаться у прямих кодах. Під час множення чисел 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 третім способом здійснюється зі старших розрядів множника, сума часткових добутків зсувається вліво, а множене нерухоме. 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 RG1(1). У результаті підсумовування вмісту RG3 і RG1 може виникнути перенос у молодший розряд регістру RG1. У старшій частині суматора, на якому здійснюється підсумовування 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 розряди добутку будуть знаходитися в регістрі RG2, а старші – в регістрі RG1. Час множення третім способом визначається аналогічно першому способу і дорівнює tm=n(tП+tЗ).

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**

***RG1(0)***

***RG1***

***n***

***0***

***RG2***

***1***

***n***

***CT***

***q***

***1***

***CT=0***

***2n+1***

***1***

***1***

***2n+1***

***SM***

***2n+1***

***n+1***

***2n+1***

***n***

**0**

***1***

***n***

***RG3***

*Рисунок 1.11-Операційна схема*

**1.3.3 Змістовний мікроалгоритм**

RG1:=0.X; RG2:=0;

RG3:=Y; CT:=15

Початок

RG1(0)

RG1.RG2:=RG1.RG2+0..0.RG3

1

RG1:=l(RG1).RG2(1)

RG2:=l(RG2).0

CT:=CT-1

CT=0

1

Кінець

*Рисунок 1.12-Змістовний мікроалгоритм*

**1.3.4 Таблиця станів регістрів**

*Таблиця 1.5- Таблиця станів регістрів*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 | СТ |
| П.С. | 001101010000111 | 0000000000000000000000000000000 | 10110101000011 | 1111 |
| 1🡨 | 011010100001110 | 0000000000000001011010100001110 |  | 1110 |
| 2 | 110101000011100 | 0000000000000010110101000011100 |  | 1101 |
| 3 | 101010000111000 | +  0000000000000000101101010000111  000000000000011100010010100011  000000000000111000100101000110 |  | 1100 |
| 4🡨 | 010100001110000 | +  0000000000000000101101010000111  000000000000111110001111001101  000000000001111100011110011010 |  | 1011 |
| 5 | 101000011100000 | 000000000011111000111100110100 |  | 1010 |
| 6 | 010000111000000 | +  0000000000000000101101010000111  000000000011111110100110111011  000000000111111101001101110110 |  | 1001 |
| 7 | 100001110000000 | 000000001111111010011011101100 |  | 1000 |
| 8🡨 | 000011100000000 | +  0000000000000000101101010000111  000000010000000000000101110011  000000100000000000001011100110 |  | 0111 |
| 9🡨 | 000111000000000 | 000001000000000000010111001100 |  | 0110 |
| 10 | 001110000000000 | 000010000000000000101110011000 |  | 0101 |
| 11 | 011100000000000 | 000100000000000001011100110000 |  | 0100 |
| 12 | 111000000000000 | 001000000000000010111001100000 |  | 0011 |
| 13🡨 | 110000000000000 | +  0000000000000000101101010000111  001000000000001000100011100111  010000000000010001000111001110 |  | 0010 |
| 14🡨 | 100000000000000 | +  0000000000000000101101010000111  010000000000010110110001010101  100000000000101101100010101010 |  | 0001 |
| 15 | 000000000000000 | +  0000000000000000101101010000111  100000000000110011001100110001 |  | 0000 |

**1.3.5 Функціональна схема з відображенням управляючих сигналів**

***0***

***n***

***RG1***

***R***

***W2***

***SL2***

***RG2***

***1***

***n***

***n***

***X***

***W1***

***SL1***

***2n+1***

***SM***

***2n+1***

***1***

***2n+1***

***1***

***n***

***n+1***

***x2***

***2n+1***

***=1***

**0**

***W3***

***CT***

***q***

***1***

***1***

***n***

***RG3***

***q***

***n***

***WCT***

***D***

***n***

***Y***

*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).RG2(1)  RG2:=l(RG2).0  CT:=CT-1 | R  W2  W3  WCT  W1  SL1  SL2  D |  | RG1(0)  CT=0 | X1  X2 |
| Початок  Кінець  R,W2,W3,WCT  W1  SL1,SL2,D  X1  X2  1  1  Z1  Z2  Z3  Z4  Z5 |

*Рисунок 1.14- Закодований мікроалгоритм*

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

Z1

-

Q1Q2

01

00

-

X1

-

Z1

-

Z3

W1

Z2

Y1,Y2,Y3

001

000

Q3Q2Q1

-

Z2

R,W2, W3,WCT

110

Z5

**-**

X2

Z4

SL1, SL2,D

010

011

*Рисунок 1.15- Граф автомата Мура*

**1.3.8 Обробка порядків і нормалізація**

.   
Знак мантиси: .

Нормалізація мантиси не потрібна.

MZ= , 1000000000001100

.

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

Зн.Р P=+1310 Зн.М M

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

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

**1.4 Четвертий спосіб множення**

**1.4.1 Теоретичне обґрунтування**

Числа множаться у прямих кодах. Під час множення чисел 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 виконується зі старших розрядів множника, сума часткових добутків залишається нерухомою, а множене зсувається вправо. 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(1), що знаходиться в старшому розряді регістру RG2, керує підсумовуванням, а в RG3 здійснюється правий зсув на один розряд, що еквівалентно множенню вмісту цього регістра на 2-1. Час виконання множення четвертим способом складає tM=ntП, визначається аналогічно до другого способу. Запишу четвертий спосіб в аналітичні формі. 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***

***2n***

***1***

***1***

***1***

***2n***

***2n***

***SM***

***2n***

***2n***

***RG2(1)***

***RG2***

***n***

***1***

***2n***

***RG3***

***2n***

***1***

*Рисунок 1.16-Операційна схема*

**1.4.3 Змістовний мікроалгоритм**

Початок

RG1:=0

RG2:=X

RG3:=0.Y

RG2(1)

1

RG1:=RG1+RG3

RG3:=0.r(RG3)

RG2:=l(RG2).0

RG2=0

1

Кінець

*Рисунок 1.17-Змістовний мікроалгоритм*

**1.4.4 Таблиця станів регістрів**

*Таблиця 1.7- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG2🡨 | RG3🡪 |
| П.С. | 0000000000000000000000000000000 | 001101010000111 | 0010110101000011100000000000000 |
| 1 | 0010110101000011100000000000000 | 011010100001110 | 0001011010100001110000000000000 |
| 2 | 0010110101000011100000000000000 | 110101000011100 | 0000101101010000111000000000000 |
| 3 | +  0000101101010000111000000000000  011100010010100011000000000000 | 101010000111000 | 0000010110101000011100000000000 |
| 4 | +  0000010110101000011100000000000  011111000111100110100000000000 | 010100001110000 | 0000001011010100001110000000000 |
| 5 | 011111000111100110100000000000 | 101000011100000 | 0000000101101010000111000000000 |
| 6 | +  0000000101101010000111000000000  011111110100110111011000000000 | 010000111000000 | 0000000010110101000011100000000 |
| 7 | 011111110100110111011000000000 | 100001110000000 | 0000000001011010100001110000000 |
| 8 | +  0000000001011010100001110000000  100000000000001011100110000000 | 000011100000000 | 0000000000101101010000111000000 |
| 9 | 100000000000001011100110000000 | 000111000000000 | 0000000000010110101000011100000 |
| 10 | 100000000000001011100110000000 | 001110000000000 | 0000000000001011010100001110000 |
| 11 | 100000000000001011100110000000 | 011100000000000 | 0000000000000101101010000111000 |
| 12 | 100000000000001011100110000000 | 111000000000000 | 0000000000000010110101000011100 |
| 13 | +  0000000000000010110101000011100  100000000000100010001110011100 | 110000000000000 | 0000000000000001011010100001110 |
| 14 | +  0000000000000001011010100001110  100000000000101101100010101010 | 100000000000000 | 0000000000000000101101010000111 |
| 15 | +  0000000000000000101101010000111  100000000000110011001100110001 | 000000000000000 | 0000000000000000010110101000011 |

**1.4.5 Функціональна схема з відображенням управляючих сигналів**

***RG2(1)***

***RG2***

***RG1***

***n***

***1***

***2n***

***1***

***2n+1***

***SL***

***W2***

***R***

***W1***

***n***

***X***

***1***

***1***

***2n***

***2n***

***SM***

***2n***

***2n***

***1***

***2n***

***RG3***

***W3***

***SR***

***n***

***1***

***Y***

***‘0’***

*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:=0.Y  RG1:=RG1+RG3  RG3:=0.r(RG3)  RG2:=l(RG2).0 | R  W2  W3  W1  SR  SL |  | RG2(1)  RG2=0 | X1  X2 |
|  |

Z1

Початок

R, W2, W3

Z2

X1

Z3

1

W1

SR,SL

Z4

X2

1

Z5

Кінець

*Рисунок 1.19- Закодований мікроалгоритм*

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

Z1

-

Q1Q2

01

00

-

X1

-

Z1

-

Z3

W1

Z4

SR,SL

Z2

Y1,Y2,Y3

001

000

Q3Q2Q1

-

Z2

R, W2, W3

110

Z5

**-**

X2

010

011

*Рисунок 1.20- Граф автомата Мура*

**1.4.8 Обробка порядків і нормалізація**

.   
Знак мантиси: .

Нормалізація мантиси не потрібна.

MZ= , 1000000000001100

.

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

Зн.Р P=+1310 Зн.М M

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

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

**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. Такий спосіб називається діленням із зсувом залишку. Черговий залишок формується в регістрі RG2 (у вихідному стані в цьому регістрі записаний Х), дільник Y знаходиться в регістрі RG1. Виходи RG2 підключені до входів SM безпосередньо, тобто ланцюги видачі коду з RG2 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. Результат формується в регістрі RG3.

**2.1.2 Операційна схема**

M

***RG1***

***1***

***n+1***

***RG2***

***n+2***

***1***

***n***

***n+2***

***SM***

***n+2***

***1***

***1***

***n+2***

***1***

***n+2***

***RG3***

*Рисунок 2.1-Операційна схема*

**2.1.3 Змістовний мікроалгоритм**

Початок

RG3:=0

RG2:=X;

RG1:=Y;

RG1:=l(RG1).

RG2:=l(RG2).0

1

RG2(1)

RG2:=RG2++1

RG2:=RG2+RG3

М

1

Кінець

*Рисунок 2.2-Змістовний мікроалгоритм*

**2.1.4 Таблиця станів регістрів**

*Таблиця 2.1- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG3 | RG2 | RG1 |
| П.С. | 0000000000000000 | 00101101010000111 | 00101101010000111 |
| 1 | 0000000000000001 | 01011010100001110  +  11101001010111101  01000011111001011 |  |
| 2 | 0000000000000011 | 10000111110010110  +  000101101010000111  10110101000011101 |  |
| 3 | 0000000000000110 | 01101010000111010  +  11101001010111101  01010011011110111 |  |
| 4 | 0000000000001101 | 10100110111101110  +  000101101010000111  11010100001110101 |  |
| 5 | 0000000000011010 | 10101000011101010  +  000101101010000111  11010101101110001 |  |
| 6 | 0000000000110100 | 10101011011100010  +  000101101010000111  11011000101101001 |  |
| 7 | 0000000001101000 | 10110001011010010  +  000101101010000111  11011110101011001 |  |
| 8 | 0000000011010000 | 10111101010110010  +  000101101010000111  11101010100111001 |  |
| 9 | 0000000110100000 | 11010101001110010  +  000101101010000111  00000010011111001 |  |
| 10 | 0000001101000001 | 00000100111110010  +  11101001010111101  11101110010101111 |  |
| 11 | 0000011010000010 | 11011100101011110  +  000101101010000111  00001001111100101 |  |
| 12 | 0000110100000101 | 00010011111001010  +  11101001010111101  11111101010000111 |  |
| 13 | 0001101000001010 | 11111010100001110  +  000101101010000111  00100111110010101 |  |
| 14 | 0011010000010101 | 01001111100101010  +  11101001010111101  00111000111100111 |  |
| 15 | 0110100000101011 | 01110001111001110  +  11101001010111101  01011011010001011 |  |
| 16 | 1101000001010111 | 10110110100010110  +  000101101010000111  11100011110011101 |  |

**2.1.5 Функціональна схема з відображенням управляючих сигналів**

M

***n+1***

***1***

***RG1***

***n+1***

***1***

***n+2***

***R***

***SL1***

***1***

***RG2***

***1***

***n+2***

***RX***

***n+2***

***W1***

***SL2***

***W3***

***W4***

*V*

MX

*X*

*S*

***1***

***1***

***n+2***

***n+2***

***SM***

***=1***

***RG3***

***W2***

***1***

***n+2***

*Y*

*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 |
| RG3:=0  RG2:=X;  RG1:=Y;  RG1:=l(RG1).  RG2:=l(RG2).0  RG2:=RG2++1  RG2:=RG2+RG1 | R  W1  W2  SL1  S2  W3  W4 |  | RG2(1)  M | | X1  X2 |
| Z1  Початок | |

R, W1, W2,V

Z2

Z3

SL1, SL2

1

X1

Z4

W3,S

Z5

W4

X2

Z6

Кінець

1

*Рисунок 2.4- Закодований мікроалгоритм*

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

Q3Q2Q1

000

Z1

-

Z2

R,W1,W2V

-

001

111

Z6

-

X2

-

Z5

W4

Z3

SL1,SL2

X2

Z4

W3,S

110

011

010

*Рисунок 2.5- Граф автомата Мура*

**2.1.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси не потрібна.

MZ= , 1101000001010111

Знак мантиси: .

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

Зн.Р P=+112 Зн.М M

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

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

**2.2 Другий спосіб ділення**

**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**

M

***2n+1***

***2n+1***

***n+1***

***1***

***RG2***

***1***

***RG1***

***1***

***p***

***2n+1***

***SM***

***2n+1***

***1***

***1***

***2n+1***

***2n+1***

***2n+1***

***1***

***RG3***

*Рисунок 2.6-Операційна схема*

**2.2.3 Змістовний мікроалгоритм**

Початок

Кінець

RG2:=RG2++1

RG3:=0.r(RG3)

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

RG2:=RG2+RG3

RG3:=0.r(RG3)

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

RG2(1)

M

1

RG1:=0;

RG2:=00.X;

RG3:=00.Y

*1*

*Рисунок 2.7-Змістовний мікроалгоритм*

**2.2.4 Таблиця станів регістрів**

*Таблиця 2.3- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 |
| П.С. | 0000000000000001 | 010110101000011100000000000000 | 0001011010100001110000000000000 |
| 1 | 0000000000000011 | 010110101000011100000000000000  +  111010010101111001000000000000  010000111110010101000000000000 | 0000101101010000111000000000000 |
| 2 | 0000000000000110 | 010000111110010101000000000000  +  111101001010111100100000000000  001110001001010001100000000000 | 0000010110101000011100000000000 |
| 3 | 0000000000001101 | 001110001001010001100000000000  +  111110100101011110010000000000  001100101110101111110000000000 | 0000001011010100001110000000000 |
| 4 | 0000000000011010 | 001100101110101111110000000000  +  111111010010101111001000000000  001100000001011110111000000000 | 0000000101101010000111000000000 |
| 5 | 0000000000110100 | 001100000001011110111000000000  +  111111101001010111100100000000  001011101010110110011100000000 | 0000000010110101000011100000000 |
| 6 | 0000000001101000 | 001011101010110110011100000000  +  111111110100101011110010000000  001011011111100010001110000000 | 0000000001011010100001110000000 |
| 7 | 0000000011010000 | 001011011111100010001110000000  +  111111111010010101111001000000  001011011001111000000111000000 | 0000000000101101010000111000000 |
| 8 | 0000000110100000 | 001011011001111000000111000000  +  111111111101001010111100100000  001011010111000011000011100000 | 0000000000010110101000011100000 |
| 9 | 0000001101000001 | 001011010111000011000011100000  +  111111111110100101011110010000  001011010101101000100001110000 | 0000000000001011010100001110000 |
| 10 | 0000011010000010 | 001011010101101000100001110000  +  111111111111010010101111001000  001011010100111011010000111000 | 0000000000000101101010000111000 |
| 11 | 0000110100000101 | 001011010100111011010000111000  +  111111111111101001010111100100  001011010100100100101000011100 | 0000000000000010110101000011100 |
| 12 | 0001101000001010 | 001011010100100100101000011100  +  111111111111110100101011110010  001011010100011001010100001110 | 0000000000000001011010100001110 |
| 13 | 0011010000010101 | 001011010100011001010100001110  +  111111111111111010010101111001  001011010100010011101010000111 | 0000000000000000101101010000111 |
| 14 | 0110100000101011 | 001011010100010011101010000111  +  111111111111111101001010111101  001011010100010000110101000100 | 0000000000000000010110101000011 |
| 15 | 1101000001010111 | 001011010100010000110101000100  +  111111111111111110100101011111  001011010100001111011010100011 | 0000000000000000001011010100001 |

**2.2.5 Функціональна схема з відображенням управляючих сигналів**

M

***2n+1***

***n+1***

***W1***

***W3***

***W4***

***R***

***SL***

***1***

***RG1***

***1***

***RG2***

***V***

MX

***X***

***1***

***S***

***SM***

***p***

***1***

***2n+1***

***2n+1***

***=1***

***W2***

***SR***

***RG3***

***2n+1***

***1***

*Y*

*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;  RG2:=00.X  RG3:=00.Y  RG2:=RG2+RG1  RG2:=RG2++1  RG1:=0.r(RG1)  RG3:=l(RG3).SM(p) | R  W1  W2  W3  W4  SR  SL |  | RG2(1)  M | X1  X2 |
|  |

Початок

Z1

Z2

R,W1,W2,V

1

X1

Z4

Z3

W3,SR,SL

W4,SR,SL ,S

X2

1

Z5

Кінець

*Рисунок 2.9- Закодований мікроалгоритм*

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

Q3Q2Q1

000

Z1

-

Z2

R,W1,W2

V

-

010

001

X2

X2

Z5

**-**

110

Z3

W4,SR,SL S

Z4

W3,SR,SL

011

*Рисунок 2.10- Граф автомата Мура*

**2.2.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси не потрібна.

MZ= , 1101000001010111

Знак мантиси: .

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

Зн.Р P=+112 Зн.М M

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

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

**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 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.

**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 |
| 0,101101010101001 | 11 | П.С. |
| 0,010110101010100 | 10 | 🡪 |
| 0,001011010101010 | 01 | 🡪 |
| 0,000101101010101 | 00 | 🡪 |

1. Додавання мантис у модифікованому ДК.

Таблиця 3.2-Додавання мантис

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

1. Нормалізація результату (В ПК).

**3.2 Операційна схема**

*m*-кількість розрядів мантиси

*n*-кількість розрядів порядку

*q*=]log2*m*[

R

L

*КС*

CT=0

*1* RG1 *m*+2 +1

1 CT *q*

*1* RG2 *n*+2

*Рисунок 3.1-Операційна схема*

Виконаємо синтез КС для визначення порушення нормалізації.

*Таблиця 3.3-Визначення порушення нормалізації*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Розряди регістру RG1 | | | 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**

Початок

CT:=*m*; RG1:=Z;

1

0

1

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

RG2:=RG2+1

1

RG1:=l(RG1).0

RG2:=RG2-1

CT:=CT-1;

Кінець

*Рисунок 3.2-Змістовний мікроалгоритм*

**3.4 Таблиця станів регістрів**

*Таблиця 3.4- Таблиця станів регістрів*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ такту** | **RG2** | **RG1** | **ЛПН(L)** | **ППН(R)** | **СT** | **Мікрооперація** |
| **ПС** | 001000 |  | 0 | 1 | 1111 |  |

**3.5 Функціональна схема з відображенням управляючих сигналів**

CT=0

R

L

Z0

SL

SR

W1,W2

1 2 3 RG1 m+2

*1* CT *q*

*=1*

*Z*

*V*

MX

W

D

*m*

*0..01*

inc

Dpz

W

1 RG2 n+2

*Pz*

*Рисунок 3.3-Функціональна схема*

**3.6 Закодований мікроалгоритм**

*Таблиця 3.5- Таблиця кодування мікрооперацій*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| CT:=m;  RG1:=Z;  Z’0 Z0:=  RG1:=RG1(1).r(RG1)  RG2:=RG2+1  RG1:=l(RG1).0  RG2:=RG2-1  CT:=CT-1; | W  W1  W2  SR  inc  SL  D2  D |  | Z’0 =0  0 | X1  X2  X3  X4 |
|  |

Z1

Початок

Z2

W,W1,V

X1

1

Z3

W2

1

SR,inc

Z4

1

Z5

SL,D2,D

X4

1

Кінець

Z6

*Рисунок 3.4- Закодований мікроалгоритм*

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

Z1

-

Q3Q2Q1

000

-

001

111

Z6

-

Z2

W,W1,V

X4

X2

Z5

SL,D,D

-

Z3

W2

110

011

Z4

SR,inc

X2

010

*Рисунок 3.5- Граф автомата Мура*

**3.8 Обробка порядків**

PZ=1000

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

Зн.Р P=+810 Зн.М M

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

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

**4. Операція добування кореня**

**Z=**

**4.1 Теоретичне обґрунтування способу**

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**

*CT=0*

***CT***

***q***

***1***

***SM***

***n+2***

***n+2***

***1***

***1***

***‘11’***

***2***

***n***

***n+2***

***1***

***2***

***2***

***RG2***

***RG3***

***RG1***

***1***

***n***

***n+2***

***n***

***1***

***1***

***1***

***n+2***

*Рисунок 4.1-Операційна схема*

**4.3 Змістовний мікроалгоритм**

Початок

RG1:=0;

RG2:=0;

RG3:=X;

CT:=15

RG2:=L2(RG2).RG3(1).RG3(2);

RG3:=L2(RG3).0.0;

1

RG2(1)

RG2:=RG2+RG1.11

RG2:=RG2+.11 RG2:=RG2+RG1.11

RG1:=L(RG1).

CT:=CT-1

CT=0

1

Кінець

*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  00000000000000010 | 101101010000111 | 1111 |
| 1 | 000000000000001 | 00000000000000010  +  11111111111111111  00000000000000001  00000000000000111 | 010110101000011 | 1110 |
| 2 | 000000000000011 | 00000000000000111  +  11111111111111011  00000000000000010  00000000000001001 | 001011010100001 | 1101 |
| 3 | 000000000000110 | 00000000000001001  +  11111111111110011  11111111111111100  11111111111110001 | 000101101010000 | 1100 |
| 4 | 000000000001101 | 11111111111110001  +  00000000000011011  00000000000001100  00000000000110000 | 000010110101000 | 1011 |
| 5 | 000000000011010 | 00000000000110000  +  11111111111001011  11111111111111011  11111111111101100 | 000001011010100 | 1010 |
| 6 | 000000000110101 | 11111111111101100  +  00000000001101011  00000000001010111  00000000101011111 | 000000101101010 | 1001 |
| 7 | 000000001101011 | 00000000101011111  +  11111111100101011  00000000010001010  00000001000101010 | 000000010110101 | 1000 |
| 8 | 000000011010111 | 00000001000101010  +  11111111001010011  00000000001111101  00000000111110100 | 000000001011010 | 0111 |
| 9 | 000000110101110 | 00000000111110100  +  11111110010100011  11111111010010111  11111101001011100 | 000000000101101 | 0110 |
| 10 | 000001101011101 | 11111101001011100  +  00000011010111011  00000000100010111  00000010001011100 | 000000000010110 | 0101 |
| 11 | 000011010111010 | 00000010001011100  +  11111001010001011  11111011011100111  11101101110011100 | 000000000001011 | 0100 |
| 12 | 000110101110100 | 11101101110011100  +  00001101011101011  11111011010000111  11101101000011100 | 000000000000101 | 0011 |
| 13 | 001101011101001 | 11101101000011100  +  00011010111010011  00000111111101111  00011111110111100 | 000000000000010 | 0010 |
| 14 | 011010111010010 | 00011111110111100  +  11001010001011011  11101010000010111  10101000001011100 | 000000000000001 | 0001 |
| 15 | 110101110100101 | 10101000001011100  +  01101011101001011  00010011110100111  01001111010011100 | 000000000000000 | 0000 |

**4.5 Функціональна схема з відображенням управляючих сигналів**

*x2*

*=1*

***1***

***WCT***

***D***

*S*

***SM***

***CT***

***q***

***1***

***n+2***

***n+2***

***1***

***1***

***q***

***n***

***‘11’***

***2***

***n***

***n+2***

*=1*

*x1*

***1***

***2***

***2***

***RG2***

***RG3***

***RG1***

***n***

***n+2***

***n***

***1***

***1***

***1***

***1***

***W1***

***R1***

***R2***

***X***

***SL2***

***SL3***

***SL1***

***W2***

***W3***

*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;  RG2:=0;  RG3:=X  CT:=15  RG2:=RG2+RG1.11  RG2:=RG2+.11  RG2:=L2(RG2).RG3(1).RG3(2)  RG3:=L2(RG3).0.0  RG1:=L(RG1).  CT:=CT-1 | R1  R2  W1  WCT  W2  W3  SL1  SL2  SL3  D |  | RG2(1)  CT=0 | X1  X2 |
|  |

Z1

Початок

Z2

R1,R2,W1,WCT

SL1,SL2

Z3

1

X1

Z5

W2

W3,S

Z4

SL3,D

Z6

X2

1

Z7

Кінець

*Рисунок 4.4- Закодований мікроалгоритм*

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

Q3Q2Q1

000

101

Z1

-

001

-

Z7

-

Z2

R1,R2,W1WCT

X2

-

100

Z3

SL1,SL2

Z6

SL3,D

011

-

111

-

Z5

W2

Z4

W3,S

110

*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**

Зн.Р P=+410 Зн.М M

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

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