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

Смоляр Андрій Григорович

Керівник: Жабін В.І.

2010 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: 922410 = 100100000010002

X2 = –10010001,0100100

Y2 = +10100,0010100001

**Виконання роботи**

***Завдання 1***

В прямому коді:

X**2:** Зн.Р P=+00010002 Зн.М M= -,100100010100100

### Y2: Зн.Р P=+00001012 Зн.М M= +,101000010100001

E=P+28

X2: Зн.Е Е=100010002 Зн.М M= -,100100010100100

### Y2: Зн.Р Е=100001012 Зн.М M= +,101000010100001

***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. | 0000000000000000 | 10010001010010**0** | 0101000010100001 | 1111 |
| 1🡪 | 0000000000000000 | 01001000101001**0** |  | 1110 |
| 2🡪 | 0000000000000000 | 00100100010100**1** | 0101000010100001 | 1101 |
| 3🡪 | 0000000000000000  +  0101000010100001  0101000010100001  0010100001010000 | 10010010001010**0** |  | 1100 |
| 4🡪 | 0001010000101000 | 01001001000101**0** |  | 1011 |
| 5🡪 | 0000101000010100 | 00100100100010**1** |  | 1010 |
| 6🡪 | 0000101000010100  +  0101000010100001  0101101010110101  0010110101011010 | 10010010010001**0** |  | 1001 |
| 7🡪 | 0001011010101101 | 01001001001000**1** |  | 1000 |
| 8🡪 | 0001011010101101  +  0101000010100001  0110011101001110  0011001110100111 | 00100100100100**0** |  | 0111 |
| 9🡪 | 0001100111010011 | 10010010010010**0** |  | 0110 |
| 10🡪 | 0000110011101001 | 11001001001001**0** |  | 0101 |
| 11🡪 | 0000011001110100 | 11100100100100**1** |  | 0100 |
| 12🡪 | 0000011001110100  +  0101000010100001  0101011100010101  0010101110001010 | 11110010010010**0** |  | 0011 |
| 13🡪 | 0001010111000101 | 01111001001001**0** |  | 0010 |
| 14🡪 | 0000101011100010 | 10111100100100**1** |  | 0001 |
| 15🡪 | 0000101011100010  +  0101000010100001  0101101110000011  0**010110111000001** | **110111100100100** |  | 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**

.

Нормалізація мантиси.

MZ= ,010110111000001 🡨 , .

,101101110000011 .

Знак мантиси: .

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

Зн.Р P=+1210 Зн.М M

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

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

**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. | 000000000000000000000000000000 | 100100010100100 | 000000000000000101000010100001 |
| 1 | 000000000000000000000000000000 | 010010001010010 | 000000000000001010000101000010 |
| 2 | 000000000000000000000000000000 | 001001000101001 | 000000000000010100001010000100 |
| 3 | 000000000000000000000000000000  +  000000000000010100001010000100  000000000000010100001010000100 | 000100100010100 | 000000000000101000010100001000 |
| 4 | 000000000000010100001010000100 | 000010010001010 | 000000000001010000101000010000 |
| 5 | 000000000000010100001010000100 | 000001001000101 | 000000000010100001010000100000 |
| 6 | 000000000000010100001010000100  +  000000000010100001010000100000  000000000010110101011010100100 | 000000100100010 | 000000000101000010100001000000 |
| 7 | 000000000010110101011010100100 | 000000010010001 | 000000001010000101000010000000 |
| 8 | 000000000010110101011010100100  +  000000001010000101000010000000  000000001100111010011100100100 | 000000001001000 | 000000010100001010000100000000 |
| 9 | 000000001100111010011100100100 | 000000000100100 | 000000101000010100001000000000 |
| 10 | 000000001100111010011100100100 | 000000000010010 | 000001010000101000010000000000 |
| 11 | 000000001100111010011100100100 | 000000000001001 | 000010100001010000100000000000 |
| 12 | 000000001100111010011100100100  +  000010100001010000100000000000  000010101110001010111100100100 | 000000000000100 | 000101000010100001000000000000 |
| 13 | 000010101110001010111100100100 | 000000000000010 | 001010000101000010000000000000 |
| 14 | 000010101110001010111100100100 | 000000000000001 | 010100001010000100000000000000 |
| 15 | 000010101110001010111100100100  +  010100001010000100000000000000  010110111000001110111100100100 | 000000000000000 | 101000010100001000000000000000 |

**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**

Нормалізація мантиси.

MZ= , 010110111000001 🡨 ,

, 101101110000011

Знак мантиси:

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

Зн.Р P=+1210 Зн.М M

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

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

**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. | 1001000101001000 | 000000000000000 | | 101000010100001 | 1111 |
| 1🡨 | 1001000101001000  0010001010010001 | +  101000010100001  101000010100001  010000101000010 | |  | 1110 |
| 2 | 0100010100100010 | 100001010000100 | |  | 1101 |
| 3 | 1000101001000101 | 000010100001000 | |  | 1100 |
| 4🡨 | 1000101001000101  0001010010001011 | +  101000010100001  101010110101001  010101101010010 | |  | 1011 |
| 5 | 0010100100010110 | 101011010100100 | |  | 1010 |
| 6 | 0101001000101101 | 010110101001000 | |  | 1001 |
| 7 | 1010010001011010 | 101101010010000 | |  | 1000 |
| 8🡨 | 1010010001011011  0100100010110110 | +  101000010100001  010101100110001  101011001100010 | |  | 0111 |
| 9🡨 | 1001000101101101 | 010110011000100 | |  | 0110 |
| 10 | 1001000101101101  0010001011011011 | +  101000010100001  111110101100101  111101011001010 | |  | 0101 |
| 11 | 0100010110110111 | 111010110010100 | |  | 0100 |
| 12 | 1000101101101111 | 110101100101000 | |  | 0011 |
| 13🡨 | 1000101101110000  0001011011100000 | +  101000010100001  011101111001001  111011110010010 | |  | 0010 |
| 14🡨 | 0010110111000001 | 110111100100100 | |  | 0001 |
| 15 | 0101101110000011 | 101111001001000 | |  | 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**

.

MZ= ,010110111000001 🡨 , ,

,101101110000011 .

Знак мантиси: .

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

Зн.Р P=+1210 Зн.М M

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

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

**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. | 000000000000000000000000000000 | 100100010100100 | 010100001010000100000000000000 |
| 1 | 000000000000000000000000000000  +  010100001010000100000000000000  010100001010000100000000000000 | 001000101001000 | 001010000101000010000000000000 |
| 2 | 010100001010000100000000000000 | 010001010010000 | 000101000010100001000000000000 |
| 3 | 010100001010000100000000000000 | 100010100100000 | 000010100001010000100000000000 |
| 4 | 010100001010000100000000000000  +  000010100001010000100000000000  010110101011010100100000000000 | 000101001000000 | 000001010000101000010000000000 |
| 5 | 010110101011010100100000000000 | 001010010000000 | 000000101000010100001000000000 |
| 6 | 010110101011010100100000000000 | 010100100000000 | 000000010100001010000100000000 |
| 7 | 010110101011010100100000000000 | 101001000000000 | 000000001010000101000010000000 |
| 8 | 010110101011010100100000000000  +  000000001010000101000010000000  010110110101011001100010000000 | 010010000000000 | 000000000101000010100001000000 |
| 9 | 010110110101011001100010000000 | 100100000000000 | 000000000010100001010000100000 |
| 10 | 010110110101011001100010000000  +  000000000010100001010000100000  010110110111111010110010100000 | 001000000000000 | 000000000001010000101000010000 |
| 11 | 010110110111111010110010100000 | 010000000000000 | 000000000000101000010100001000 |
| 12 | 010110110111111010110010100000 | 100000000000000 | 000000000000010100001010000100 |
| 13 | 010110110111111010110010100000  +  000000000000010100001010000100  010110111000001110111100100100 | 000000000000000 | 000000000000001010000101000010 |

**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**

.

MZ= , 010110111000001 🡨 , ;

,101101110000011 .

Знак мантиси: .

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

Зн.Р P=+1210 Зн.М M

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1, | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 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. 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 | 00100100010100100 | 00101000010100001 |
| 1 | 000000000000000**1** | 01001000101001000  +  11010111101011111  00100000010100111 |  |
| 2 | 00000000000000**1**1 | 01000000101001110  +  11010111101011111  00011000010101101 |  |
| 3 | 0000000000000**1**11 | 00110000101011010  +  11010111101011111  00001000010111001 |  |
| 4 | 000000000000**1**111 | 00010000101110010  +  11010111101011111  11101000011010001 |  |
| 5 | 00000000000**1**1110 | 11010000110100010  +  00101000010100001  11111001001000011 |  |
| 6 | 0000000000**1**11100 | 11110010010000110  +  00101000010100001  00011010100100111 | 00101000010100001 |
| 7 | 000000000**1**111001 | 00110101001001110  +  11010111101011111  00001100110101101 |  |
| 8 | 00000000**1**1110011 | 00011001101011010  +  11010111101011111  11110001010111001 |  |
| 9 | 0000000**1**11100110 | 11100010101110010  +  00101000010100001  00001011000010011 |  |
| 10 | 000000**1**111001101 | 00010110000100110  +  11010111101011111  11101101110000101 |  |
| 11 | 00000**1**1110011010 | 11011011100001010  +  00101000010100001  00000011110101011 |  |
| 12 | 0000**1**11100110101 | 00000111101010110  +  11010111101011111  11011111010110101 |  |
| 13 | 000**1**111001101010 | 10111110101101010  +  00101000010100001  11100111000001011 |  |
| 14 | 00**1**1110011010100 | 11001110000010110  +  00101000010100001  11110110010110111 |  |
| 15 | 0**1**11100110101000 | 11101100101101110  +  00101000010100001  00010101000001111 |  |
| 16 | **1**,111001101010001 | 00101010000011110  +  11010111101011111  00000001101111101 |  |

**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= , 111001101010001

Знак мантиси: .

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

Зн.Р P=+1210 Зн.М M

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1, | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 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. | 000000000000000**1** | 0100100010100100000000000000000 | 0010100001010000100000000000000 |
| 1 | 00000000000000**1**1 | +  1101011110101111100000000000000  0010000001010011100000000000000 | 0001010000101000010000000000000 |
| 2 | 0000000000000**1**11 | 0010000001010011100000000000000  +  1110101111010111110000000000000  0000110000101011010000000000000 | 0000101000010100001000000000000 |
| 3 | 000000000000**1**111 | 0000110000101011010000000000000  +  1111010111101011111000000000000  0000001000010111001000000000000 | 0000010100001010000100000000000 |
| 4 | 00000000000**1**1110 | 0000001000010111001000000000000  +  1111101011110101111100000000000  1111110100001101000100000000000 | 0000001010000101000010000000000 |
| 5 | 0000000000**1**11100 | 1111110100001101000100000000000  +  0000001010000101000010000000000  1111111110010010000110000000000 | 0000000101000010100001000000000 |
| 6 | 000000000**1**111001 | 1111111110010010000110000000000  +  0000000101000010100001000000000  0000000011010100100111000000000 | 0000000010100001010000100000000 |
| 7 | 00000000**1**1110011 | 0000000011010100100111000000000  +  1111111101011110101111100000000  0000000000110011010110100000000 | 0000000001010000101000010000000 |
| 8 | 0000000**1**11100110 | 0000000000110011010110100000000  +  1111111110101111010111110000000  1111111111100010101110010000000 | 0000000000101000010100001000000 |
| 9 | 000000**1**111001101 | 1111111111100010101110010000000  +  0000000000101000010100001000000  0000000000001011000010011000000 | 0000000000010100001010000100000 |
| 10 | 00000**1**1110011010 | 0000000000001011000010011000000  +  1111111111101011110101111100000  1111111111110110111000010100000 | 0000000000001010000101000010000 |
| 11 | 0000**1**11100110101 | 1111111111110110111000010100000  +  0000000000001010000101000010000  0000000000000000111101010110000 | 0000000000000101000010100001000 |
| 12 | 000**1**111001101010 | 0000000000000000111101010110000  +  1111111111111010111101011111000  1111111111111011111010110101000 | 0000000000000010100001010000100 |
| 13 | 00**1**1110011010100 | 1111111111111011111010110101000  +  0000000000000010100001010000100  1111111111111110011100000101100 | 0000000000000001010000101000010 |
| 14 | 0**1**11100110101000 | 1111111111111110011100000101100  +  0000000000000001010000101000010  1111111111111111101100101101110 | 0000000000000000101000010100001 |
| 15 | **1**111001101010001 | 1111111111111111101100101101110  +  0000000000000000101000010100001  0000000000000000010101000001111 | 0000000000000000010100001010000 |

**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

-

110

010

001

X2

X2

Z5

**-**

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= ,111001101010001

Знак мантиси:

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

Зн.Р P=+310 Зн.М M

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1, | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 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 |
| 0,101000010100001 | 11 | П.С. |
| 0,010100001010000 | 10 | 🡪 |
| 0,001010000101000 | 01 | 🡪 |
| 0,000101000010100 | 00 | 🡪 |

1. Додавання мантис у модифікованому ДК.

Таблиця 3.2-Додавання мантис

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

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 |  |
| **1** | 000111 |  | 0 | 0 | 1110 | Z’0 Z0:=  RG1:=l(RG1).0  RG2:=RG2-1  CT:=CT-1 |

**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=111.

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

Зн.Р P=+710 Зн.М M

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1, | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 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. | 0000000000000000 | 000000000000000000 | 0100100010100100 | 10000 |
| 1 | 0000000000000001 | 000000000000000001  +  111111111111111111  000000000000000000 | 0010001010010000 | 1111 |
| 2 | 0000000000000010 | 000000000000000000  +  111111111111111011  111111111111111011 | 1000101001000000 | 1110 |
| 3 | 0000000000000100 | 111111111111101110  +  000000000000001011  111111111111111001 | 0010100100000000 | 1101 |
| 4 | 0000000000001000 | 111111111111100100  +  000000000000010011  111111111111110111 | 1010010000000000 | 1100 |
| 5 | 0000000000010001 | 111111111111011110  +  000000000000100011  000000000000000001 | 1001000000000000 | 1011 |
| 6 | 0000000000100010 | 000000000000000110  +  111111111110111011  111111111111000001 | 0100000000000000 | 1010 |
| 7 | 0000000001000100 | 111111111100000101  +  000000000010001011  111111111110010000 | 0000000000000000 | 1001 |
| 8 | 0000000010001000 | 111111111001000000  +  000000000100010011  111111111101010011 | 0000000000000000 | 1000 |
| 9 | 0000000100010000 | 111111110101001100  +  000000001000100011  111111111101101111 | 0000000000000000 | 0111 |
| 10 | 0000001000100001 | 111111110110111100  +  000000010001000011  000000000111111111 | 0000000000000000 | 0110 |
| 11 | 0000010001000010 | 000000011111111100  +  111111011101111011  111111111101110111 | 0000000000000000 | 0101 |
| 12 | 0000100010000101 | 111111110111011100  +  000001000100001011  000000111011100111 | 0000000000000000 | 0100 |
| 13 | 0001000100001011 | 000011101110011100  +  111101110111101011  000001100110000111 | 0000000000000000 | 0011 |
| 14 | 0010001000010111 | 000110011000011100  +  111011101111010011  000010000111101111 | 0000000000000000 | 0010 |
| 15 | 0100010000101110 | 001000011110111100  +  110111011110100011  111111111101011111 | 0000000000000000 | 0001 |
| 16 | 1000100001011101 | 111111110101111100  +  010001000010111011  010000111000110111 | 0000000000000000 | 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 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |

**5. Синтез управляючого автомату для операційного пристрою множення першим способом.**

x3x2x1 +1=001=1 – операція множення першим способом.

x2x1=00 – RS-тригери.

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

За закодованим мікроалгоритмом (Рис. 1.4) складемо таблицю:

*Таблиця 5.1 Таблиця кодування сигналів*

|  |  |
| --- | --- |
| Входи операційного автомата | Виходи управляючого автомата |
| R, W2, W3, WCT | Y1 |
| W1 | Y2 |
| SR1,SR2,D | Y3 |

**5.2 Мікроалгоритм в термінах управляючого автомата**

Зробимо автомат Мура циклічним задля зменшення кількості вершин.

Початок

Z1

Y1

Z2

X1

1

Z3

Y2

Z4

Y3

X2

1

Z1

Кінець

*Рисунок 5.1- Закодований мікроалгоритм*

Будуємо граф автомата Мура

00

01

Q1Q2

-

Z1

-

Z2

Y1

X1

X2

-

Z4

Y3

Z3

Y2

10

11

*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 JK-тригера.

*Таблиця 5.2-Структурна таблиця автомата*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Перехід | Q2Q1 | Q2Q1 | x1x2 | y1y2y3 | R2 | S2 | R1 | S1 |
| *z1z2* | 00 | 01 | -- | 000 | - | 0 | 0 | 1 |
| *z2 z3* | 01 | 11 | 1- | 100 | 0 | 1 | 0 | - |
| *z2 z4* | 01 | 10 | 0- | 100 | 0 | 1 | 1 | 0 |
| *z3 z4* | 11 | 10 | -- | 010 | 0 | - | 1 | 0 |
| *z4 z3* | 10 | 11 | 10 | 001 | 0 | - | 0 | 1 |
| *z4 z1* | 10 | 00 | -1 | 001 | 1 | 0 | - | 0 |
| *z4 z4* | 10 | 10 | 00 | 001 | 0 | - | - | 0 |

RS:

**5.4 Синтез функцій виходів і переходів**

0

1

0

0

Y3

1

0

0

0

Y2

0

0

1

0

Q2

Q1

Y1

0

0

0

0

0

0

0

1

-

-

1

1

0

0

1

1

Q2

Q1

X2

X1

S1

1-

1

-

-

1

1-

-

0

0

0

0

0

1

1

0

0

Q2

Q1

X2

X1

R1

-

-

0

-

-

-

-

0

-

1

1

0

0

1

1

0

0

Q2

Q1

X2

X1

S2

0

0

1

0

0

0

1

0

0

0

-

-

0

0

-

-

Q2

Q1

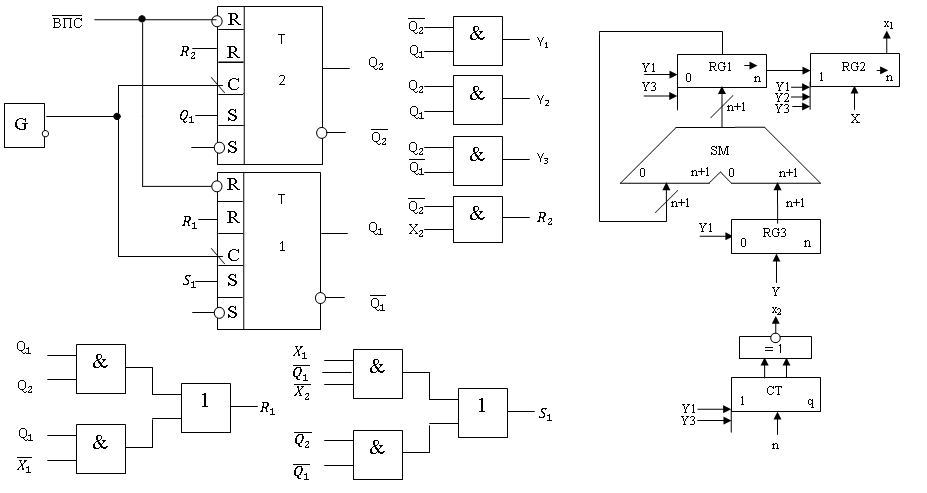
X2

X1

R2

*Рисунок 5.3- Діаграми Вейча*

­­

**

*Рисунок 5.5- Функціональна схема пристрою*

**Висновки:** Таким чином, виконуючи дану розрахункову роботу, я повторив такі операції над числами в двійковому коді , як множення, ділення, добування кореня та додавання. Для кожної операції була подана операційна схема, змістовний мікроалгоритм, таблиця станів регістрів, функціональна схема з відображенням управляючих сигналів, граф відповідного закодованого мікроалгоритму і ,власне, закодований мікроалгоритм. Був синтезований управляючий автомат для операційного пристрою множення першим способом, згідно з варіантом, відповідно, даний операційний пристрій був побудований на RS-тригерах.