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-64

Бандурін Владислав Юрійович

Керівник: Жабін В.І.

Київ

2017р.

**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: 640210 = 11001000000102

 і ,

X2 = –001000,0110

Y2 = +110000,0010

**Виконання роботи**

***Завдання 1***

В прямому коді:

X2пк = 1.001000,0110

Y2пк = 0.110000,0010

*X2* у формі з плаваючою комою (з порядком та мантисою):

*P*=610=0,1102 *M*=1, 001000

Зн.РZ PZ=+610 Зн.МZ MZ

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

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

*X2* = 0,110 1,001000

*Y2* у формі з плаваючою комою (з порядком та мантисою):

P=610=0,1102 *M*=0,110000

*Y2*=0,110 0,110000

Зн.РZ PZ=+610 Зн.МZ MZ

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

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

Запишемо X та Y за стандартом ANSI/IEEE 754-2008:

Px =127+6 = 133 = 100001012

Х:

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

Порядок Мантиса

PY =127+6 = 133 = 100001012

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

Порядок Мантиса

2. **Операції над числами**

2.1 **Множення першим способом**

2.1.1 **Теоретичне обґрунтування способу**

Числа множаться у прямих кодах, знакові та основні розряди обробляються окремо. Для визначення знака добутку здійснюють підсумування по модулю 2 цифр, що розміщуються в знакових розрядах співмножників.

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

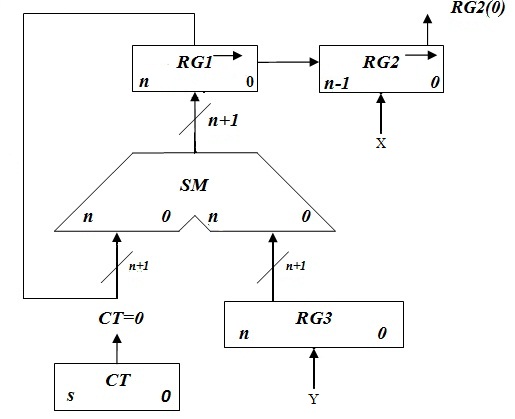
що рівносильно

.

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

.

2.1.2 **Операційна схема.**

****

*Рисунок 2.1.1. Операційна схема пристрою множення першим способом*

2.1.3 **Змістовний мікроалгоритм**

Початок

Кінець

RG1:=0

RG2:=X

RG3:=Y

CT:=n

RG1:=RG1+RG3

RG1:=0.r(RG1)

RG2:=RG1[0].r(RG2)

CT:=CT-1

RG2[0]

CT=0

1

1

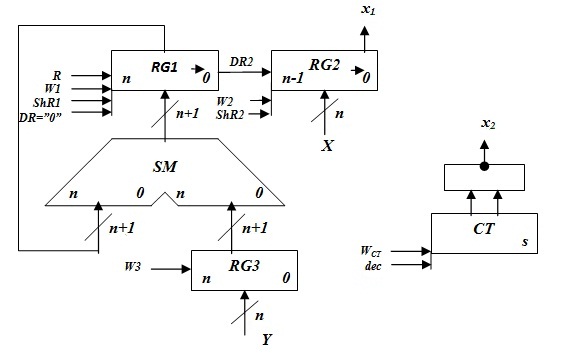
*Рисунок 2.1.2. Змістовний мікроалгоритм пристрою множення першим способом*

2.1.4 **Таблиця станів регістрів**

*Таблиця 2.1.1. Таблиця станів регістрів пристрою множення першим способом*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 | СТ |
| П.С. | 000000 | 00100**0** | 110000 | 110 |
| 1 | 000000 | 00010**0** | 110000 | 101 |
| 2 | 000000 | 00001**0** | 110000 | 100 |
| 3 | 000000 | 00000**1** | 110000 | 011 |
| 4 | +110000  110000  011000 | 00000**0** | 110000 | 010 |
| 5 | 001100 | 00000**0** | 110000 | 001 |
| 6 | 0**00110** | **000000** | 110000 | 000 |

2.1.5 **Функціональна схема з відображенням управляючих сигналів**



*Рисунок 2.1.3. Функціональна схема пристрою множення першим способом*

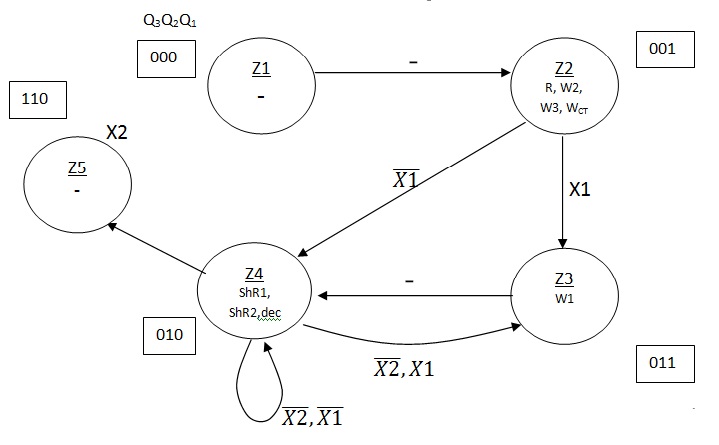
2.1.5 **Закодований мікроалгоритм**

*Таблиця 2.1.2. Таблиця кодування операцій і логічних умов пристрою множення першим способом*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| G1:=0  RG2:=X  RG3:=Y  CT:=15  RG1:=RG1+RG3  RG1:=0.r(RG1)  RG2:=RG1[0].r(RG2)  CT:=CT-1 | R  W2  W3  WCT  W1  ShR1  ShR2  dec  Початок  Кінець  R, W2, W3, WCT  W1  ShR1,ShR2,dec  X1  X2  1  1  Z1  Z2  Z3  Z4  Z5 |  | RG2[0]  CT=0 | X1  X2 |
|
|
|  |

*Рисунок 2.1.5. Закодований мікроалгоритм пристрою множення першим способом*

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

**

*Рисунок 2.1.6. Граф автомата Мура пристрою множення першим способом*

2.5.7 **Обробка порядків**

Нормалізація мантиси:

MZ = ,001100 <= ; ;

,110000 ; .

Знак мантиси:

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

Зн.РZ PZ=+1010 Зн.МZ MZ

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

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

2.2 **Другий спосіб множення**

2.2.1 **Теоретичне обґрунтування способу**

Числа множаться у прямих кодах, знакові та основні розряди обробляються окремо. Визначення знака добутку здійснюють підсумування по модулю 2 цифр, що розміщуються в знакових розрядах співмножників.

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

Вираз

подамо у вигляді

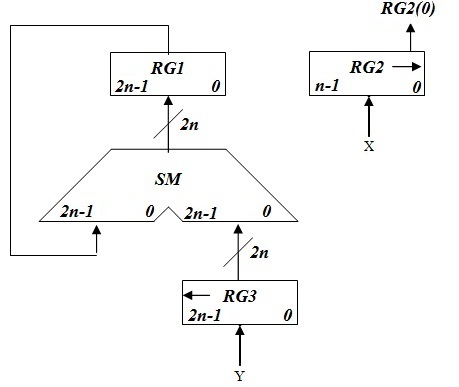
.

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

.

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

2.2.2 **Операційна схема**



*Рисунок 2.2.1. Операційна схема пристрою множення другим способом*

*2.2.3.*  **Змістовний мікроалгоритм**

Початок

Кінець

RG1:=0

RG2:=X

RG3:=00.Y

RG1:=RG1+RG3

RG2:=0.r(RG2)

RG3:=l(RG3).0

RG2[0]

RG2=0

1

1

*Рисунок 2.2.2. Змістовний мікроалгоритм пристрою множення другим способом*

2.2.4. **Таблиця станів регістрів**

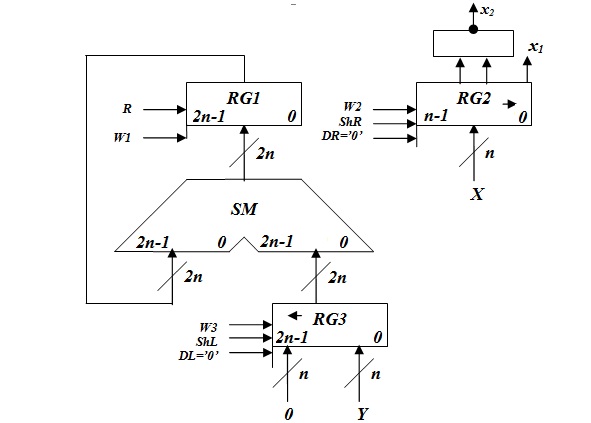
X **=** 0,1010 1,110000

Y = 0,0110 0,110000

*Таблиця 2.2.1. Таблиця станів регістрів пристрою множення другим способом*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1 | RG2 🡪 | RG3 🡨 | C |
| П.С. | 000000000000 | 11000**0** | 000000110000 | 0 |
| 1 | 000000000000 | 01100**0** | 000001100000 | 0 |
| 2 | 000000000000 | 00110**0** | 000011000000 | 0 |
| 3 | 000000000000 | 00011**0** | 000110000000 | 0 |
| 4 | 000000000000 | 00001**1** | 001100000000 | 0 |
| 5 | +001100000000  001100000000 | 00000**1** | 011000000000 | 0 |
| 6 | +011000000000  100100000000 | **000000** | 110000000000 | 0 |

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



*Рисунок 2.2.3. Функціональна схема пристрою множення другим способом*

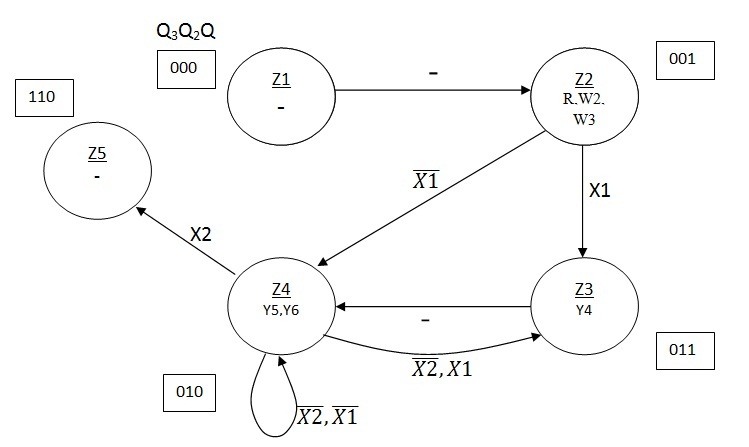
2.2.6 **Закодований мікроалгоритм**

*Таблиця 2.2.2. Таблиця кодування мікрооперацій пристрою множення другим способом*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| RG1:=0  RG2:=X  RG3:=Y  RG1:=RG1+RG3  RG2:=0.r(PG2)  RG3:=l(RG3).0 | R  W2  W3  W1  ShR  ShL |  | RG2[0]  RG2=0 | X1  X2 |
| Початок  Кінець  R,W2,W3  W1  ShR, ShL  X1  X2  1  1  Z1  Z2  Z3  Z4  Z5 |

*Рисунок 2.2.4. Закодований мікроалгоритм множення пристрою множення другим способом*

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

****

*Рисунок 2.2.5. Граф автомата Мура пристрою множення другим способом*

2.2.8 **Обробка порядків**

MZ= ,100100

Нормалізація мантиси не потрібна

Знак мантиси: .

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

Зн.РZ PZ=+1610 Зн.МZ MZ

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

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

2.3 **Третій спосіб множення**

2.3.1 **Теоретичне обґрунтування способу**

Числа множаться у прямих кодах, знакові та основні розряди обробляються окремо. Визначення знака добутку здійснюють підсумування по модулю 2 цифр, що розміщуються в знакових розрядах співмножників.

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

Вираз

подамо у вигляді

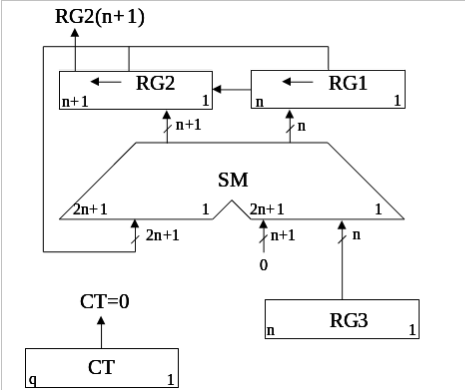
*.*

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

.

з початковими умовами *Z0=0, i=1.*

2.3.2 **Операційна схема**



*Рисунок 2.3.1. Операційна схема пристрою множення третім способом*

2.3.3 **Змістовний мікроалгоритм**

Початок

Кінець

RG1:=0; RG2:=X;

RG3:=Y; CT:=n

RG1:=RG1+RG3

RG1:=l(RG1).0

RG2:=l(RG2).0

CT:=CT-1

RG2[n-1]

CT=0

1

1

*Рисунок 2.3.2. Змістовний мікроалгоритм пристрою множення третім способом*

2.3.4 **Таблиця станів регістрів**

*Таблиця 2.3.1. Таблиця станів регістрів пристрою множення третім способом*

X **=** 0,10000 1,100100

Y = 0,0110 0,110000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | RG1 | RG3 | RG2 | CT |
| ПС | 00000000000 | 110000 | **1**00100.0 | 110 |
| 1 | +000000110000  000000110000  000001100000 | 110000 | **0**010000 | 101 |
| 2 |  | 110000 | **0**100000 | 100 |
| 3 |  | 110000 | **1**000000 | 011 |
| 4 |  | 110000 | **0**000000 | 010 |
| 5 |  | 110000 | **0**000000 | 001 |
| 6 |  | 110000 | **0**000000 | 000 |

**2.3.5 Обробка порядків і нормалізація**

.

MZ= ,

Нормалізація мантиси не потрібна.

Знак мантиси: .

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

Зн.РZ PZ=+2110 Зн.МZ MZ

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

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

**1.4 Четвертий спосіб множення**

**1.4.1 Теоретичне обґрунтування**

Числа множаться у прямих кодах, знакові та основні розряди обробляються окремо. Визначення знака добутку здійснюють підсумування по модулю 2 цифр, що розміщуються в знакових розрядах співмножників.

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

Вираз

*подамо у вигляді*

*.*

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

.

з початковими значеннями i=1, Y0=2-1Y, Z0=0.

**1.4.2 Операційна схема**

***RG1***

***0***

***2n-1***

***2n-1***

***2n-1***

***0***

***0***

***SM***

***2n***

***2n***

***RG2(n-1)***

***RG2***

***0***

***n-1***

***2n***

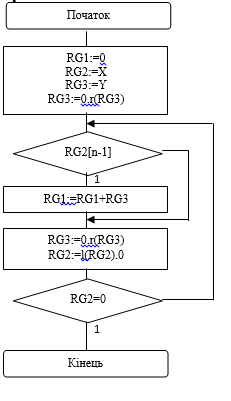
***RG3***

***0***

***2n-1***

*Рисунок 1.16-Операційна схема*

**1.4.3 Змістовний мікроалгоритм**



*Рисунок 1.17-Змістовний мікроалгоритм*

**1.4.4 Таблиця станів регістрів**

X **=** 0,10101 1,110110

Y = 0,00110 0,110000

*Таблиця 1.7- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 |
| П.С. | 000000000000 | 110110 | 011000000000 |
| 1 | 011000000000 | 101100 | 001100000000 |
| 2 | +001100000000  100100000000 | 011000 | 000110000000 |
| 3 | 100100000000 | 110000 | 000011000000 |
| 4 | +000011000000  100111000000 | 100000 | 000001100000 |
| 5 | +000001100000  **101000100000** | **000000** | 000000110000 |

Округлення результату для його запису в задану розрядну мантису:

**101000100000**

+1

***101001000000***

*Mz=****1****,* ***101001***

**2.4.5 Обробка порядків і нормалізація**

.

*Mz=****1****,* ***101001***

Нормалізація мантиси не потрібна.

Знак мантиси: .

Зн.РZ PZ=+2810 Зн.МZ MZ

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

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

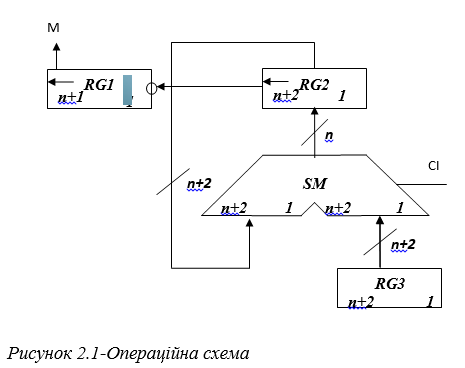
**2.5 Перший спосіб ділення**

**2.5.1 Теоретичне обґрунтування**

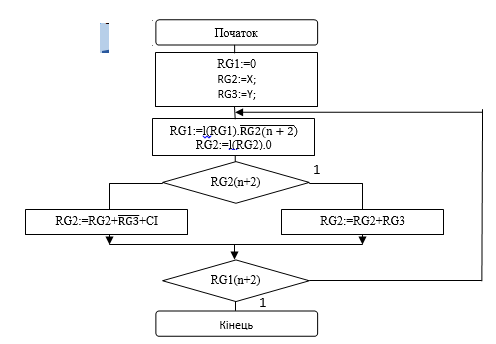
Нехай ділене Х і дільник Y є n-розрядними правильними дробами, поданими в прямому коді. В цьому випадку знакові й основні розряди операндів обробляються окремо. Знак результату визначається шляхом підсумовування по модулю 2 цифр, записаних в знакових розрядах.

При реалізації ділення за першим методом здійснюється зсув вліво залишку при нерухомому дільнику. Черговий залишок формується в регістрі RG2 (у вихідному стані в цьому регістрі записаний Х). Виходи RG2 підключені до входів СМ безпосередньо, тобто ланцюги видачі коду з RG2 не потрібні. Час для підключення n+1 цифри частки визначається виразом t=(n+1)(tд+tз), де tд - тривалість виконання мікрооперації додавання-віднімання; tз - тривалість виконання мікрооперації зсуву.

**2.5.2 Операційна схема**



**2.5.3 Змістовний мікроалгоритм**



*Рисунок 2.5-Змістовний мікроалгоритм*

**2.5.4 Таблиця станів регістрів**

Так як Mx<My виконується, тому мантису діленого не зсуваємо вправо.

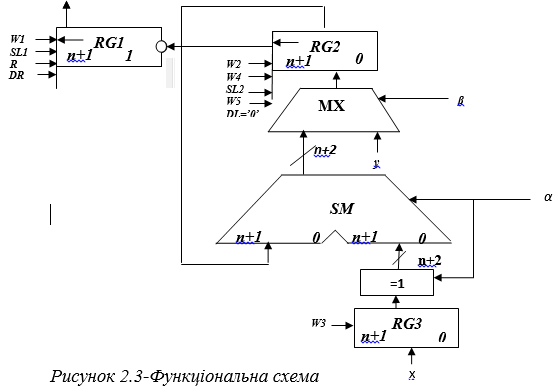
*M=1, 101001*

Py= 610 = 0,1102 *My*=0,110000

*Таблиця 2.1- Таблиця станів регістрів*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 | МО |
| П.С. | 0000000 | 00101001 | 00110000 |  |
| 1 | 0000001 | 01010010  +11010000  00100010 |  | 🡨 RG2🡨RG1  RG2:=RG2-RG3 |
| 2 | 0000011 | 01000100  +11010000  00010100 |  | 🡨 RG2🡨RG1  RG2:=RG2-RG3 |
| 3 | 0000111 | 00101000  +11010000  11111000 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 4 | 0001110 | 11110000  +00110000  00100000 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 5 | 0011101 | 01000000  +11010000  00010000 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 6 | 0111011 | 00100000  +11010000  11110000 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |
| 7 | 1**110110** | 11100000  +00110000  00010000 |  | 🡨 RG2🡨RG1  RG2:=RG2+RG3 |

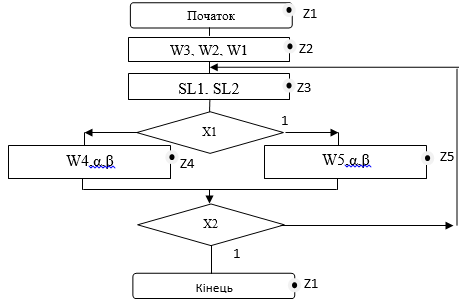
**2.5.5 Функціональна схема з відображенням управляючих сигналів**



**2.5.6 Закодований мікроалгоритм**

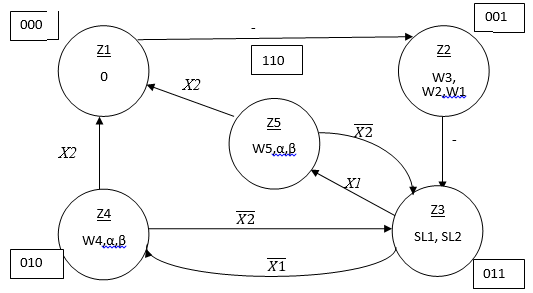
*Таблиця 2.2- Таблиця кодування мікрооперацій*

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



*Рисунок 2.4- Закодований мікроалгоритм*

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



*Рисунок 2.5- Граф автомата Мура*

**2.5.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси не потрібна.

Знак мантиси: .

Зн.РZ PZ=+2210 Зн.МZ MZ

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

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

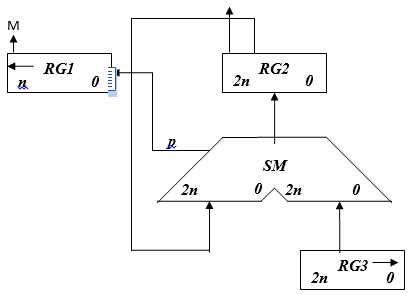
**2.6 Другий спосіб ділення(без відновлення остачі)**

**2.6.1 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.

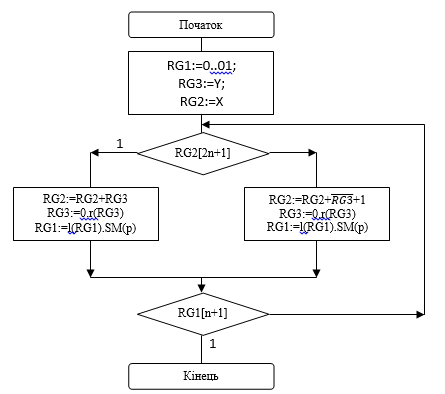
Остача нерухома, дільник зсувається праворуч. Як і при множенні з нерухомою сумою часткових добутків можна водночас виконувати підсумування і віднімання, зсув в регістрах Y,Z. Тобто 1 цикл може складатися з 1 такту, це дає прискорення відносно 1-го способу.

**2.6.2 Операційна схема**



*Рисунок 2.6-Операційна схема*

**2.6.3 Змістовний мікроалгоритм**



*Рисунок 2.7-Змістовний мікроалгоритм*

**2.2.4 Таблиця станів регістрів**

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

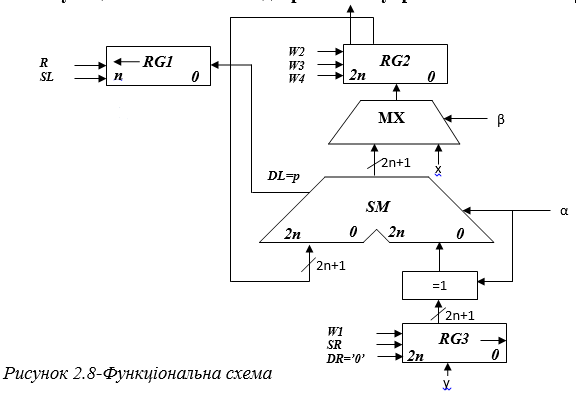
Px=22(10)+1(10)=23(10)=0. (2) Mx = 1.011011

Py= 610 = 0,1102 *My*=0,110000

*Таблиця 2.3- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 |
| П.С. | 0000001 | 001101100000 | 001100000000 |
| 1 | 0000011 | 001101100000  +110100000000  000001100000 | 000110000000 |
| 2 | 0000110 | 000001100000  +111010000000  111011100000 | 000011000000 |
| 3 | 0001100 | 111011100000  + 000011000000  111110100000 | 000001100000 |
| 4 | 0011001 | 111110100000  + 000001100000  000000000000 | 000000110000 |
| 5 | 0110010 | 000000000000  +111111010000  111111010000 | 000000011000 |
| 6 | 1**100100** | 111111010000  +000000011000  111111101000 | 000000001100 |

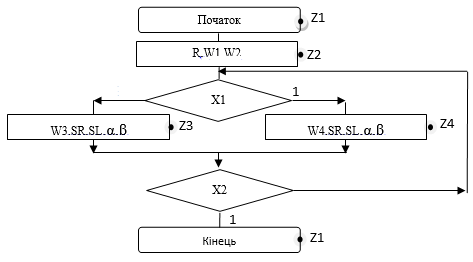
**2.6.5 Функціональна схема з відображенням управляючих сигналів**



**2.6.6 Закодований мікроалгоритм**

*Таблиця 2.4- Таблиця кодування мікрооперацій*

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



*Рисунок 2.9- Закодований мікроалгоритм*

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

-

Z1

0

Z2

R, W1, W2

Z4

W4, SR, SL,α,β

Z3

W3, SR, SL,α,β

*X2*

*X1*

*X2*

00

01

11

10

*Рисунок 2.10- Граф автомата Мура*

**2.6.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси не потрібна.

Знак мантиси: .

MZ=1,100100

Зн.РZ PZ=+1710 Зн.МZ MZ

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

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

**3. Операція додавання чисел**

**Z=X+Y.**

**3.1 Теоретичне обґрунтування способу**

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

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

1. Порівняння порядків.

Px =1710 =2 Mх=1,100100

Py= 610 = 0,1102 *My*=0,110000

,

.

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 |
| 0,110000 | 1011 | П.С. |
| 0,011000 | 1010 | 🡪 |
| 0,001100 | 1001 | 🡪 |
| 0,000110 | 1000 | 🡪 |
| 0,000011 | 0111 | 🡪 |
| 0,000001 | 0110 | 🡪 |
| 0,000000 | 0101 | 🡪 |
| 0,000000 | 0100 | 🡪 |
| 0,000000 | 0011 | 🡪 |
| 0,000000 | 0010 | 🡪 |
| 0,000000 | 0001 | 🡪 |
| 0,000000 | 0000 | 🡪 |

1. Додавання мантис у модифікованому ДК.

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

Рисунок 3.1-Додавання мантис

**3.2 Операційна схема**

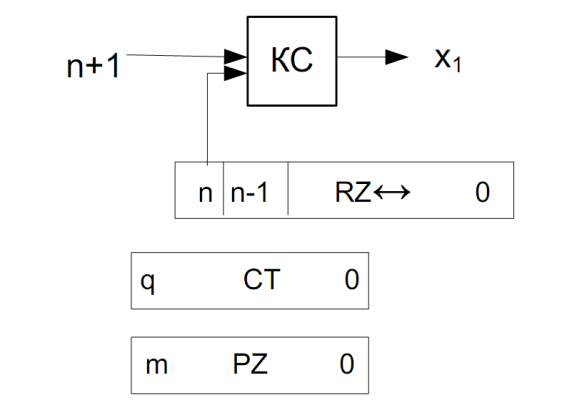
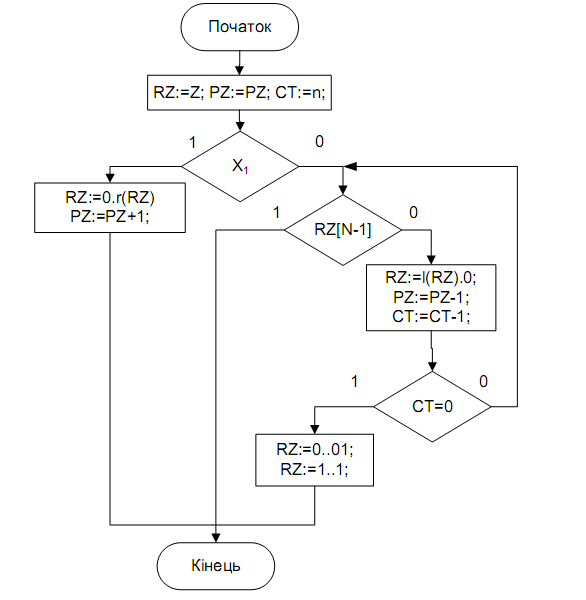
****

Рисунок 3.1-Операційна схема.

**3.3 Змістовний мікроалгоритм**

****

*Рисунок 3.2-Змістовний мікроалгоритм*

**3.4 Таблиця станів регістрів**

*Таблиця 3.4- Таблиця станів регістрів*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ такту** | **RGPZ** | **RGZ** | **ЛПН(L)** | **ППН(R)** | **СT** |
| **ПС** | 101 | 11, | 0 | 0 | 100 |

**3.5 Функціональна схема з відображенням управляючих сигналів**

R

L

Z0

ShL

ShR

W1,W2

m+1 m m-1 RGZ 0

q CT 1

CT=0

inc

dec

W

W

dec

n+1 RGPZ 0

*Рисунок 3.5Функціональна схема*

**3.6 Закодований мікроалгоритм**

*Таблиця 3.4- Таблиця кодування мікрооперацій*

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

Z1

Початок

Z2

W,W1

X1

1

Z3

W2

1

ShR,inc

Z4

1

Z5

ShL,dec,dec

X4

1

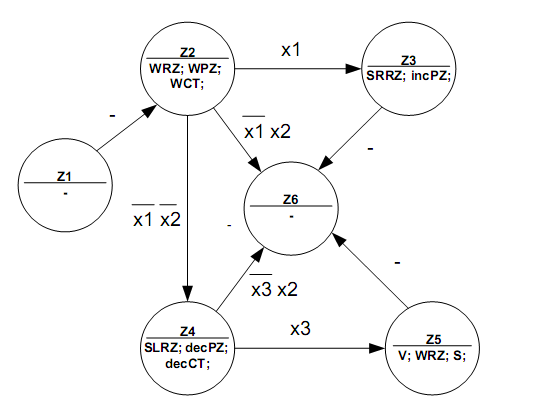
Кінець

Z6

*Рисунок 3.6- Закодований мікроалгоритм*

|  |
| --- |
|  |

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

****

*Рисунок 3.5- Граф автомата Мура*

**3.8 Обробка порядків**

PZ=.

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

Зн.Р P Зн.М M

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

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

**0.10001 1.100100**

**4. Операція добування кореня**

**Z=**

**4.1 Теоретичне обґрунтування способу**

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

; ;

.

Виконання операції зводиться до послідовності дій:

1. Одержання остачі. ;

2. Якщо , то .

3. Якщо , то .

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

**4.2 Операційна схема**

🡨

RG1

🡨

n+2

1

=1

RG2

RG3

🡨

n

1

n

1

SM

n+2

1

n+2

1

“1”

2

n+2

n

n+2

CT

s

1

“q”

2

2

*Рисунок 4.1-Операційна схема*

**4.3 Змістовний мікроалгоритм**

Початок

RG3:=X;

RG2:=0..0;

RG1:=0..0

CT:=6

Кінець

RG2:=RG2+RG1.11

RG2:=RG2+.11

RG2[n+2]

1

CT=0

1

RG1:=L(RG1).

CT:=CT-1

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

RG3:=L2[RG3].0.0

*Рисунок 4.2-Змістовний мікроалгоритм*

**4.4 Таблиця станів регістрів**

Px =1810 =100102 Mх=0,010010

*Таблиця 4.1- Таблиця станів регістрів*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 | CT |
| П.С. | 000000 | 00000000 | 010010 | 110 |
| 1🡨  🡨 | 000001 | 00000001  +11111111  00000000 | 001000 | 101 |
| 2🡨  🡨 | 000010 | 00000000  +11111011  11111011 | 100000 | 100 |
| 3🡨  🡨 | 000100 | 11101110  +00001011  11111001 | 000000 | 011 |
| 4🡨  🡨 | 001000 | 11100100  +00010011  11110111 | 000000 | 010 |
| 5🡨  🡨 | 010000 | 11011100  +00100011  11111111 | 000000 | 001 |
| 6🡨  🡨 | 100001 | 11111100  +01000011  00111111 | 000000 | 000 |

**Обробка порядків**

PZ=PX/2=18/2=910=10012.

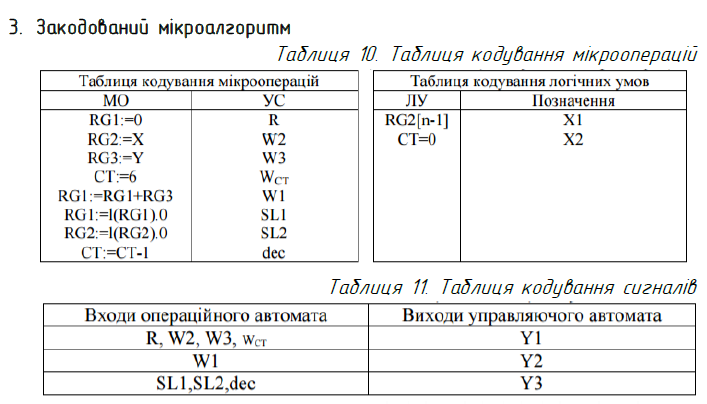
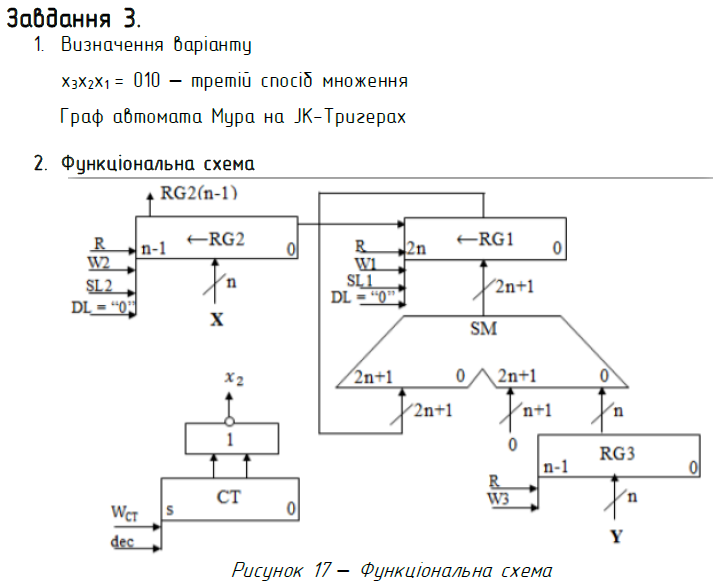
**4.9 Форма запису нормалізованого результату з плаваючою комою в пам’ять**

Pz=0.1001 Mz=0, 100001

Зн.Р P Зн.М M

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

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



****

Рисунок 18 *–* Блок схема автомата

**4. Граф автомата Мура з кодами вершин**

****

Рисунок 19 *–* Граф автомата

**5. Структурна таблиця автомата**

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

Таблиця 12 – Таблиця переходів

****

6. Синтез функцій виходів і переходів

****

****

****

****

****

****

****

****

Рисунок 20. Функціональна схема управляючого автомата. Автомат

**Висновок**

У даній розрахунковій роботі було виконано операції з числами в двійковому коді з плаваючою комою, а саме: множення чотирма способами, ділення двома способами, додавання та знаходження кореня. Було побудувано управляючий автомат Мура на тригерах JK і елементах булевого базиса для операції множення третім способом.

Зроблено мінімізацію функцій тригерів і в середовищі AFDK побудована функціональна схема автомата. Під час виконання даної розрахункової роботи було повторено матеріал курсу «Комп’ютерна логіка - 1», а також закріплено знання з курсу «Комп’ютерна логіка - 2».

Було використано наступну літературу:

1) Жабін В.І., Жуков І.А., Клименко І.А.,Ткаченко В.В. Прикладна теорія цифрових автоматів: Навчальний посібник.–К.: Книжкове вид-во НАУ, 2009. – 360 с.

2) Конспект лекцій з курсу «Комп’ютерна логіка - 1»

3) Конспект лекцій з курсу «Комп’ютерна логіка – 2».