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

Попадюк Віталій Васильович

Керівник: Жабін В.І.

2010 р.

ЗМІСТ

Завдання та обґрунтування варіанту……………………………………………5

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

1.1 Перший спосіб множення……………………………………………….7

1.1.1 Теоретичне обґрунтування способу……………………7

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’ять…………………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……………………………….…..11

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

1.2.5 Функціональна схема пристрою…………………………….12

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

1.2.7 Граф управляючого автомата………………………………..13

1.2.8 Обробка порядків ………………………………………………….14

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………………………………...…..16

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 ………………………………………….…...…….20

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

1.4 Четвертий спосіб множення………………...…………………………..…..21

1.4.1 Теоретичне обґрунтування способу………………………...21

1.4.2 Операційна схема……………………………………………….…….21

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

1.4.4 Таблиця станів регістрів……………………………….…………..22

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

1.4.6 Закодований мікроалгоритм………………………….…………23

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………………………………………………….……...25

2.1 Перший спосіб………………………………………………………….…25

2.1.1 Теоретичне обґрунтування способу………………………25

2.1.2 Операційна схема……………………………………………….…….25

2.1.3 Змістовний мікроалгоритм……………………….……………..26

2.1.4 Таблиця станів регістрів………………………………….………26

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

2.1.6 Закодований мікроалгоритм…………………………………..28

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

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

2.1.9 Форма подання результату в пам’ятті…………………....28

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

2.2.1 Теоретичне обґрунтування способу…………………..…..29

2.2.2 Операційна схема…………………………………………….………29

2.2.3 Змістовний мікроалгоритм…………………………………….30

2.2.4 Таблиця станів регістрів………………………………….……….30

2.2.5 Функціональна схема пристрою………………………………31

2.2.6 Закодований мікроалгоритм…………………………………...32

2.2.7 Граф управляючого автомата………………………………….32

2.2.8 Обробка порядків …………………………………………………...33

2.2.9 Форма подання результату в пам’ятті………….………...33

3 Операція додавання чисел

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

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

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

3.4 Таблиця станів регістрів…………………………………….……….35

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…………………...…...37

4 Операція додавання чисел

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

4.2 Операційна схема……………………………………………….…………38

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

4.4 Таблиця станів регістрів……………………………………………….39

4.5 Функціональна схема пристрою……………………………………40

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

4.7 Граф управляючого автомата………………………….…………….41

4.8 Обробка порядків ……………………………………………..……………41

4.9 Форма подання результату в пам’ятті…………………………...41

5 Синтез управляючого автомата для операційного пристрою (дати назву пристрою)

5.1 Таблиця співвідношення управляючих входів операційного автомата і виходів управляючого автомата ………………………....42

5.2 Мікроалгоритм в термінах управляючого автомата ……..42

5.3 Структурна таблиця автомата………………………………………...43

5.4 Синтех функцій виходів переходів………….………………………43

5.5 Функціональна схема пристою (виходи управляючого автомата підключені до входів операційного автомата)……....45

**Обґрунтування варіанту**

Номер залікової книжки: 922110 = 100100000001012

X2 = –10010001,0001101

Y2 = +10100,0010000001

**Виконання роботи**

***Завдання 1***

В прямому коді:

**X2:**

P = +810 ­­= 0.00010002

E = 100010002

M = 0.1001000100011012

### Y2:

P = +510 = 0.00001012

E = 100001012

M = 0.1010000100000012

***Завдання 2***

**1. Операція множення чисел**.

Z=XY, де X-множене, Y-множник.

**1.1 Перший спосіб множення.**

**1.1.1 Теоретичне обгрунтування.**

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

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

що рівносильно

.

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

.

**1.1.2 Операційна схема.**

***RG2(0)***

***RG2***

***RG1***

***0***

***n***

***n-1***

***0***

***n+1***

***n***

***n***

***0***

***0***

***SM***

***n+1***

***n+1***

***0***

***RG3***

***n***

***CT***

***0***

***n***

***CT=0***

*Рисунок 1.1.2- Операційна схема.*

**1.1.3 Змістовний мікроалгоритм.**

Початок

Кінець

RG1:=0

RG2:=Y

RG3:=X

CT:=n

RG1:=RG1+RG3

RG1:=0.r(RG1)

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

CT:=CT-1

RG2[]

CT=0

1

1

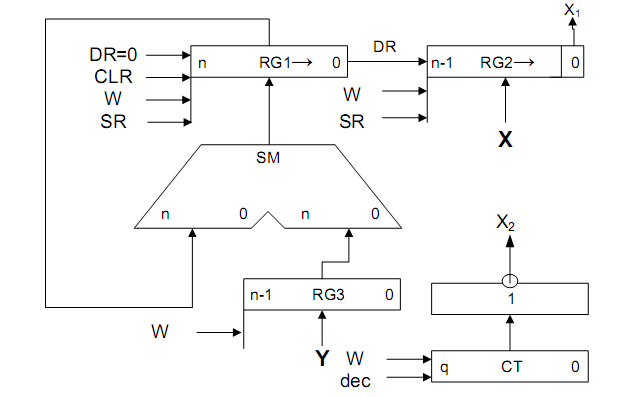
*Рисунок 1.1.3- Змістовний мікроалгоритм.*

**1.1.4 Таблиця станів регістрів.**

|  |  |  |  |
| --- | --- | --- | --- |
| RG1 | RG2 | RG3 | CT |
| 0000000000000000 | 101000010000001 | 1001000100011010 | 15 |
| 0000000000000000 1001000100011010  1001000100011010 | 010100001000000 | 1001000100011010 | 14 |
| 0100100010001101 | 101010000100000 | 1001000100011010 | 13 |
| 0010010001000110 | 010101000010000 | 1001000100011010 | 12 |
| 0001001000100011 | 101010100001000 | 1001000100011010 | 11 |
| 0000100100010001 | 110101010000100 | 1001000100011010 | 10 |
| 0000010010001000 | 011010101000010 | 1001000100011010 | 9 |
| 0000001001000100 | 001101010100001 | 1001000100011010 | 8 |
| 0000000100100010 1001000100011010  1001001000111100 | 000110101010000 | 1001000100011010 | 7 |
| 0100100100011110 | 000011010101000 | 1001000100011010 | 6 |
| 0010010010001111 | 100001101010100 | 1001000100011010 | 5 |
| 0001001001000111 | 110000110101010 | 1001000100011010 | 4 |
| 0000100100100011 | 111000011010101 | 1001000100011010 | 3 |
| 0000010010010001 1001000100011010  1001010110101011 | 111100001101010 | 1001000100011010 | 2 |
| 0100101011010101 | 111110000110101 | 1001000100011010 | 1 |
| 0010010101101010 1001000100011010  1011011010000100 | 011111000011010 | 1001000100011010 | 0 |

*Таблиця 1.1-Таблиця станів регістрів.*

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

****

*Рисунок 1.1.5- Функціональна схема.*

**1.1.6 Закодований мікроалгоритм.**

*Таблиця 1.1.6-Таблиця кодування операцій і логічних умов.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| G1:=0  RG2:=Y  RG3:=X  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 |  | RG2[0]  CT=0 | X1  X2 |
|
|
|  |

Початок

Z2

R, W2, W3, WCT

1

X1

Z3

W1

X2

ShR1,ShR2,dec

1

Кінець

Z5

Z1

Z4

*Рисунок 1.1.6-Закодований мікроалгоритм.*

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

Z2

R, W2, W3, WCT

Z3

W1

Z4

ShR1, ShR2,dec

Z1

-

-

X1

-

X2

000

001

011

010

Q3Q2Q1

Z5

**-**

110

*Рисунок 1.1.7-Граф автомата Мура*

**1.1.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси.

MZ= , 0101101101000010 🡨 , .

, 101101101000010 .

Знак мантиси: .

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

Зн.Р P Зн.М M

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

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

**1.2 Другий спосіб множення**

**1.2.1 Теоретичне обгрунтування**

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

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

Вираз

подамо у вигляді

.

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

.

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

**1.2.2 Операційна схема**

***RG2(0)***

***RG1***

***0***

***2n-1***

***n-1***

***0***

***RG2***

***2n***

***2n-1***

***2n-1***

***0***

***0***

***SM***

***2n***

***RG3***

***0***

***2n-1***

*Рисунок 1.2.2-Операційна схема*

**1.2.3 Змістовний мікроалгоритм**

Початок

RG1:=0

RG2:=X

RG3:=Y

RG2[0]

1

RG1:=RG1+RG3

RG2:=0.r(RG2)

RG3:=l(RG3).0

RG2=0

1

Кінець

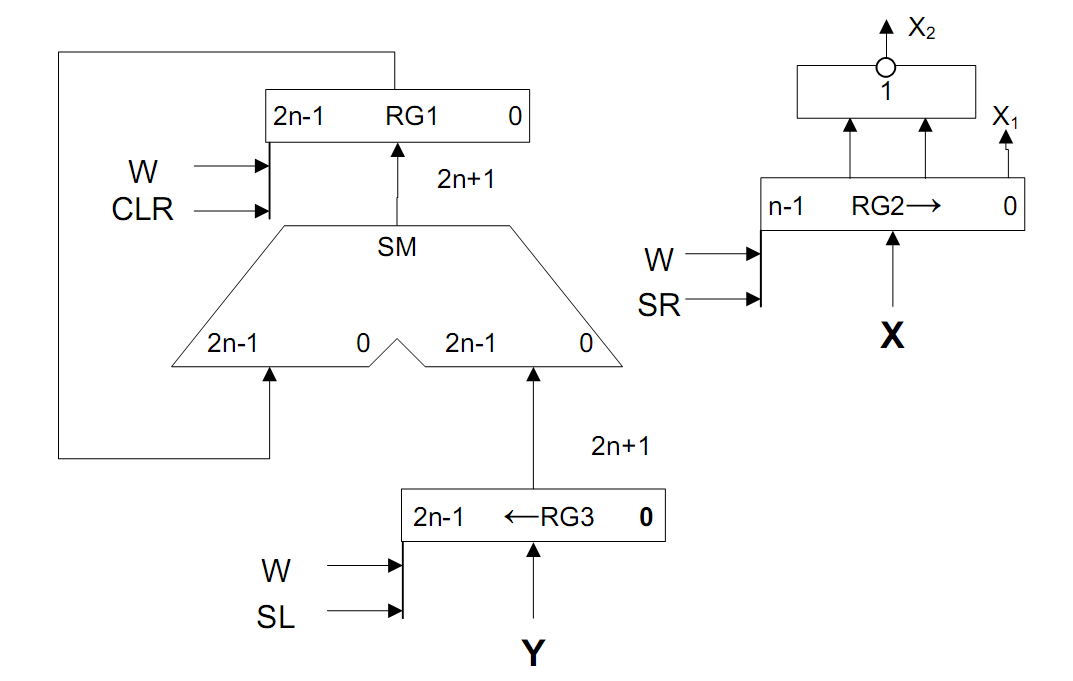
*Рисунок 1.7-Змістовний мікроалгоритм*

**1.2.4 Таблиця станів регістрів**

*Таблиця 1.2.4- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| RG1 | RG2 | RG3 | CT |
| 000000000000000000000000000000 | 101000010000001 | 000000000000000100100010001101 | 15 |
| 000000000000000000000000000000 000000000000000100100010001101  000000000000000100100010001101 | 010100001000000 | 0  00000000000001001000100011010 | 14 |
| 000000000000000100100010001101 | 001010000100000 | 000000000000010010001000110100 | 13 |
| 000000000000000100100010001101 | 000101000010000 | 000000000000100100010001101000 | 12 |
| 000000000000000100100010001101 | 000010100001000 | 000000000001001000100011010000 | 11 |
| 000000000000000100100010001101 | 000001010000100 | 000000000010010001000110100000 | 10 |
| 000000000000000100100010001101 | 000000101000010 | 000000000100100010001101000000 | 9 |
| 000000000000000100100010001101 | 000000010100001 | 000000001001000100011010000000 | 8 |
| 000000000000000100100010001101 000000001001000100011010000000  000000001001001000111100001101 | 000000001010000 | 000000010010001000110100000000 | 7 |
| 000000001001001000111100001101 | 000000000101000 | 000000100100010001101000000000 | 6 |
| 000000001001001000111100001101 | 000000000010100 | 000001001000100011010000000000 | 5 |
| 000000001001001000111100001101 | 000000000001010 | 000010010001000110100000000000 | 4 |
| 000000001001001000111100001101 | 000000000000101 | 000100100010001101000000000000 | 3 |
| 000000001001001000111100001101 000100100010001101000000000000  000100101011010101111100001101 | 000000000000010 | 001001000100011010000000000000 | 2 |
| 000100101011010101111100001101 | 000000000000001 | 010010001000110100000000000000 | 1 |
| 000100101011010101111100001101 010010001000110100000000000000  010110110100001001111100001101 | 000000000000000 | 100100010001101000000000000000 | 0 |

**1.2.5 Функціональна схема з відображенням управляючих сигналів**

****

*Рисунок 1.2.5- Функціональна схема*

**1.2.6 Закодований мікроалгоритм**

*Таблиця 1.2.6 – Таблиця кодування мікрооперацій.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| RG1:=0  RG2:=X  RG3:=Y  RG1:=RG1+RG3  RG2:=0.r(PG2)  RG3:=l(RG3).0 | R  W2  W3  W1  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 1.2.6- Закодований мікроалгоритм.*

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

Z2

Y1,Y2,Y3

Z3

Y4

Z4

Y5,Y6

Z1

-

-

X1

-

00

01

Z2

R,W2, W3

Z1

-

-

X2

000

001

011

010

Q3Q2Q1

Z5

**-**

110

*Рисунок 1.2.7- Граф автомата Мура*

**1.2.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси.

MZ= , 0101101101000010 🡨 , .

, 101101101000010 .

Знак мантиси: .

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

Зн.Р P Зн.М M

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

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

**Третій спосіб множення**

**1.3.1 Теоретичне обгрунтування**

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

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

Вираз

*подамо у вигляді*

*.*

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

.

з початковими умовами *Z0=0, i=1.*

**1.3.2 Операційна схем**

***RG2(n)***

***0***

***n-1***

***RG1***

***CT***

***0***

***q***

***CT=0***

***RG2***

***0***

***n***

***n+1***

***n***

***2n***

***2n***

***0***

***0***

***SM***

***n+1***

***2n+1***

***n***

***n-1***

***0***

***RG3***

**0**

***n-1***

***0***

*Рисунок 1.3.2-Операційна схем*

**1.3.3 Змістовний мікроалгоритм**

RG1:=0; RG2:=Y;

RG3:=X; CT:=15

Початок

RG2[n]

RG1:=RG1+RG3

1

CT=0

Кінець

1

RG2:=l(RG2).RG1[n-1]

RG1:=l(RG1).0

CT:=CT-1

*Рисунок 1.3.3-Змістовний мікроалгоритм*

**1.3.4 Таблиця станів регістрів**

*Таблиця 1.3.4- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| RG2 | RG1 | RG3 | CT |
| 0101000010000001 | 000000000000000 | 100100010001101 | 16 |
| 1010000100000010 | 000000000000000 | 100100010001101 | 15 |
| 1010000100000010 | 000000000000000 100100010001101  100100010001101 | 100100010001101 | 14 |
| 0100001000000101 | 001000100011010 |  |  |
| 1000010000001010 | 010001000110100 | 100100010001101 | 13 |
| 1000010000001010 | 010001000110100 100100010001101  110101011000001 | 100100010001101 | 12 |
| 0000100000010101 | 101010110000010 |  |  |
| 0001000000101011 | 010101100000100 | 100100010001101 | 11 |
| 0010000001010110 | 101011000001000 | 100100010001101 | 10 |
| 0100000010101101 | 010110000010000 | 100100010001101 | 9 |
| 1000000101011010 | 101100000100000 | 100100010001101 | 8 |
| 1000000101011010 | 101100000100000 100100010001101  010000010101101 | 100100010001101 | 7 |
| 0000001010110110 | 100000101011010 |  |  |
| 0000010101101101 | 000001010110100 | 100100010001101 | 6 |
| 0000101011011010 | 000010101101000 | 100100010001101 | 5 |
| 0001010110110100 | 000101011010000 | 100100010001101 | 4 |
| 0010101101101000 | 001010110100000 | 100100010001101 | 3 |
| 0101011011010000 | 010101101000000 | 100100010001101 | 2 |
| 1010110110100000 | 101011010000000 | 100100010001101 | 1 |
| 1010110110100000 | 101011010000000 100100010001101  001111100001101 | 100100010001101 | 0 |
| 0101101101000010 | 011111000011010 |  |  |

**1.3.5 Функціональна схема з відображенням управляючих *сигналів***

X1

DL

***R***

***W***

***ShL***

***DL=’0’***

***RG1***

***n-1***

***0***

***RG2***

***0***

***n***

***W***

***SL***

***DL=’0’***

***n+1***

***n***

***CT***

***0***

**q**

1

***W***

***dec***

***X2***

V

MX

Y

***0***

***0***

***2n***

***2n***

***SM***

***n+1***

***2n+1***

***n***

***RG3***

**0**

***n-1***

***0***

***n***

***Y***

*Рисунок 1.13-Функціональна схема*

**1.3.6 Закодований мікроалгоритм**

*Таблиця 1.3.6- Таблиця кодування мікрооперацій*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| RG1:=0  RG2:=X  RG3:=Y  CT:=15  RG1:=RG1+RG3  RG1:=l(RG1).0  RG2:=l(RG2).RG1[n-1]  CT:=CT-1 | R  W2  W3  WCT  W1  ShL1  ShL2  dec |  | RG2[n]  CT=0 | X1  X2 |
| Початок  Кінець  R,W2,W3,WCT  W1  ShL1,ShL2,dec  X1  X2  1  1  Z1  Z2  Z3  Z4  Z5 |

*Рисунок 1.3.6- 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

ShL1, ShL2,dec

010

011

*Рисунок 1.3.7- Граф автомата Мура*

**1.3.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси.

MZ= , 0101101101000010 🡨 , .

, 101101101000010 .

Знак мантиси: .

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

Зн.Р P Зн.М M

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

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

***2n***

***2n***

***0***

***0***

***SM***

***2n+1***

***2n+1***

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

***RG2***

***0***

***n-1***

***2n+1***

***RG3***

***0***

***2n***

*Рисунок 1.4.2-Операційна схема*

**1.4.3 Змістовний мікроалгоритм**

Початок

RG1:=0

RG2:=Y

RG3:=X

RG3:=0.r(RG3)

RG2[n-1]

1

RG1:=RG1+RG3

RG3:=0.r(RG3)

RG2:=l(RG2).0

RG2=0

1

Кінець

*Рисунок 1.4.3-Змістовний мікроалгоритм*

**1.4.4 Таблиця станів регістрів**

*Таблиця 1.4.4- Таблиця станів регістрів*

|  |  |  |
| --- | --- | --- |
| RG1 | RG2 | RG3 |
| 0000000000000000000000000000000 | 101000010000001 | 0100100010001101000000000000000 |
| 0000000000000000000000000000000 0100100010001101000000000000000  0100100010001101000000000000000 | 010000100000010 | 0010010001000110100000000000000 |
| 0100100010001101000000000000000 | 100001000000100 | 0001001000100011010000000000000 |
| 0100100010001101000000000000000 0001001000100011010000000000000  0101101010110000010000000000000 | 000010000001000 | 0000100100010001101000000000000 |
| 0101101010110000010000000000000 | 000100000010000 | 0000010010001000110100000000000 |
| 0101101010110000010000000000000 | 001000000100000 | 0000001001000100011010000000000 |
| 0101101010110000010000000000000 | 010000001000000 | 0000000100100010001101000000000 |
| 0101101010110000010000000000000 | 100000010000000 | 0000000010010001000110100000000 |
| 0101101010110000010000000000000 0000000010010001000110100000000  0101101101000001010110100000000 | 000000100000000 | 0000000001001000100011010000000 |
| 0101101101000001010110100000000 | 000001000000000 | 0000000000100100010001101000000 |
| 0101101101000001010110100000000 | 000010000000000 | 0000000000010010001000110100000 |
| 0101101101000001010110100000000 | 000100000000000 | 0000000000001001000100011010000 |
| 0101101101000001010110100000000 | 001000000000000 | 0000000000000100100010001101000 |
| 0101101101000001010110100000000 | 010000000000000 | 0000000000000010010001000110100 |
| 0101101101000001010110100000000 | 100000000000000 | 0000000000000001001000100011010 |
| 0101101101000001010110100000000 0000000000000001001000100011010  0101101101000010011111000011010 | 000000000000000 | 0000000000000000100100010001101 |

**1.4.5 Функціональна схема з відображенням управляючих сигналів**

***X2***

1

***RG1***

***0***

***2n***

***CLR***

***W3***

***X1***

***RG2***

***X***

***n-1***

***0***

***2n+1***

***SL***

***W1***

***2n***

***2n***

***0***

***0***

***SM***

***2n+1***

***2n+1***

***2n***

***0***

***RG3***

***W2***

***SR***

***Y***

*Рисунок 1.4.5-Функціональна схема*

**1.4.6 Закодований мікроалгоритм**

*Таблиця 1.4.6- Таблиця кодування мікрооперацій*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| RG1:=0  RG2:=Y  RG3:=X  RG1:=RG1+RG3  RG3:=0.r(RG3)  RG2:=l(RG2).0 | R  W2  W3  W1  ShR  ShL |  | RG2[n-1]  RG2=0 | X1  X2 |
|  |

Z1

Початок

R, W2, W3, ShR

Z2

X1

Z3

1

W1

ShR,ShL

Z4

X2

1

Z5

Кінець

*Рисунок 1.4.6- Закодований мікроалгоритм*

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

Z1

-

Q1Q2

01

00

-

X1

-

Z1

-

Z3

W1

Z4

ShR,ShL

Z2

Y1,Y2,Y3

001

000

Q3Q2Q1

-

Z2

R, W2, W3, ShR

110

Z5

**-**

X2

010

011

*Рисунок 1.4.7- Граф автомата Мура*

**1.4.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси.

MZ= , 0101101101000010 🡨 , .

, 101101101000010 .

Знак мантиси: .

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

Зн.Р P Зн.М M

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0. | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 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. Черговий залишок формується в регістрі Р2 (у вихідному стані в цьому регістрі записаний Х). Виходи Р2 підключені до входів СМ безпосередньо, тобто ланцюги видачі коду з Р2 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.

**2.1.2 Операційна схема**

***RG2(n+1)***

***RG3(n)***

***RG3***

***n***

***0***

***RG2***

***0***

***n+1***

***n+2***

***SM***

***0***

***0***

***n+2***

***n+1***

***n+1***

***n+1***

***0***

***RG1***

*Рисунок 2.1.2-Операційна схема*

**2.1.3 Змістовний мікроалгоритм**

Початок

RG3:=0

RG2:=00.X;

RG1:=00.Y;

RG3[n]

1

RG2[n+1]

1

RG2:=RG2+RG1

RG3:=l(RG3).

RG2:=l(RG2).0

RG2:=RG2++1

Кінець

*Рисунок 2.1.3-Змістовний мікроалгоритм*

**2.1.4 Таблиця станів регістрів**

*Таблиця 2.1.4- Таблиця станів регістрів*

|  |  |  |
| --- | --- | --- |
| RG3 | RG2 | RG1 |
| 0000000000000000 | 00100100010001101 | 00101000010000001 |
| 0000000000000001 | 01001000100011010 11010111101111111  00100000010011001 | 00101000010000001 |
| 0000000000000011 | 01000000100110010 11010111101111111  00011000010110001 | 00101000010000001 |
| 0000000000000111 | 00110000101100010 11010111101111111  00001000011100001 | 00101000010000001 |
| 0000000000001111 | 00010000111000010 11010111101111111  11101000101000001 | 00101000010000001 |
| 0000000000011110 | 11010001010000010 00101000010000001  11111001100000011 | 00101000010000001 |
| 0000000000111100 | 11110011000000110 00101000010000001  00011011010000111 | 00101000010000001 |
| 0000000001111001 | 00110110100001110 11010111101111111  00001110010001101 | 00101000010000001 |
| 0000000011110011 | 00011100100011010 11010111101111111  11110100010011001 | 00101000010000001 |
| 0000000111100110 | 11101000100110010 00101000010000001  00010000110110011 | 00101000010000001 |
| 0000001111001101 | 00100001101100110 11010111101111111  11111001011100101 | 00101000010000001 |
| 0000011110011010 | 11110010111001010 00101000010000001  00011011001001011 | 00101000010000001 |
| 0000111100110101 | 00110110010010110 11010111101111111  00001110000010101 | 00101000010000001 |
| 0001111001101011 | 00011100000101010 11010111101111111  11110011110101001 | 00101000010000001 |
| 0011110011010110 | 11100111101010010 00101000010000001  00001111111010011 | 00101000010000001 |
| 0111100110101101 | 00011111110100110 11010111101111111  11110111100100101 | 00101000010000001 |
| 1111001101011010 | 11101111001001010 00101000010000001  00010111011001011 | 00101000010000001 |

**2.1.5 Функціональна схема з відображенням управляючих сигналів**

X1

X2

***RZ***

***1***

***n+1***

***0***

***n***

***RG3***

***DL***

***W3***

***ShL1***

***0***

***n+1***

***RG2***

***1***

***n+2***

***RX***

***n+2***

MX

Y

***W2***

***W4***

***ShL2***

***W5***

***DL=’0’***

d

V

***0***

***SM***

***n+1***

***n+1***

***0***

***n+2***

***n+2***

=1

***W1***

***RG1***

***0***

***n+1***

***X1***

*Рисунок 2.1.5-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;  RG3:=l(RG3).  RG2:=l(RG2).0  RG2:=RG2++1  RG2:=RG2+RG1 | W3  W2  W1  ShL1  ShL2  W4  W5 |  | RG2[n+1]  RG3[n] | | X1  X2 |
| Z1  Початок | |

W3, W2, W1

Z2

Z3

ShL1, ShL2

1

X1

Z4

W4

Z5

W5

X2

Z6

Кінець

1

*Рисунок 2.1.6- Закодований мікроалгоритм*

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

Q3Q2Q1

000

Z1

-

-

001

111

Z6

-

Z2

Y1,Y2,Y3

X2

-

Z5

**-**

Z3

Y4

X2

Z4

Y5,Y6

110

011

010

*Рисунок 2.1.7- Граф автомата Мура*

**2.1.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси не потрібна.

MZ= , 111001101011010

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 | 1 | 0 | 1 | 0 |

**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**

***RG3(n)***

***RG2(2n)***

***0***

***0***

***n***

***RG3***

***2n***

***RG2***

***p***

***SM***

***2n***

***0***

***2n***

***0***

***2n+1***

***2n+1***

***2n+1***

***2n***

***0***

***RG1***

*Рисунок 2.2.2-Операційна схема*

**2.2.3 Змістовний мікроалгоритм**

Початок

Кінець

RG2:=RG2+RG1

RG1:=0.r(RG1)

RG3:=l(RG3).SM(p)

RG2:=RG2++1

RG1:=0.r(RG1)

RG3:=l(RG3).SM(p)

RG2[2n]

RG3[n]

1

1

RG3:=00..1;

RG1:=0.Y;

RG2:=00.X

*Рисунок 2.2.3-Змістовний мікроалгоритм*

**2.2.4 Таблиця станів регістрів**

*Таблиця 2.2.4- Таблиця станів регістрів*

|  |  |  |
| --- | --- | --- |
| RG3 | RG2 | RG1 |
| 0000000000000001 | 0010010001000110100000000000000 | 0010100001000000100000000000000 |
| 0000000000000001 | 0010010001000110100000000000000 1101011110111111100000000000000  1111110000000110000000000000000 | 0001010000100000010000000000000 |
| 0000000000000001 | 1111110000000110000000000000000 0001010000100000010000000000000  0001000000100110010000000000000 | 0000101000010000001000000000000 |
| 0000000000000011 | 0001000000100110010000000000000 1111010111101111111000000000000  0000011000010110001000000000000 | 0000010100001000000100000000000 |
| 0000000000000111 | 0000011000010110001000000000000 1111101011110111111100000000000  0000000100001110000100000000000 | 0000001010000100000010000000000 |
| 0000000000001111 | 0000000100001110000100000000000 1111110101111011111110000000000  1111111010001010000010000000000 | 0000000101000010000001000000000 |
| 0000000000011110 | 1111111010001010000010000000000 0000000101000010000001000000000  1111111111001100000011000000000 | 0000000010100001000000100000000 |
| 0000000000111100 | 1111111111001100000011000000000 0000000010100001000000100000000  0000000001101101000011100000000 | 0000000001010000100000010000000 |
| 0000000001111001 | 0000000001101101000011100000000 1111111110101111011111110000000  0000000000011100100011010000000 | 0000000000101000010000001000000 |
| 0000000011110011 | 0000000000011100100011010000000 1111111111010111101111111000000  1111111111110100010011001000000 | 0000000000010100001000000100000 |
| 0000000111100110 | 1111111111110100010011001000000 0000000000010100001000000100000  0000000000001000011011001100000 | 0000000000001010000100000010000 |
| 0000001111001101 | 0000000000001000011011001100000 1111111111110101111011111110000  1111111111111110010111001010000 | 0000000000000101000010000001000 |
| 0000011110011010 | 1111111111111110010111001010000 0000000000000101000010000001000  0000000000000011011001001011000 | 0000000000000010100001000000100 |
| 0000111100110101 | 0000000000000011011001001011000 1111111111111101011110111111100  0000000000000000111000001010100 | 0000000000000001010000100000010 |
| 0001111001101011 | 0000000000000000111000001010100 1111111111111110101111011111110  1111111111111111100111101010010 | 0000000000000000101000010000001 |
| 0011110011010110 | 1111111111111111100111101010010 0000000000000000101000010000001  0000000000000000001111111010011 | 0000000000000000010100001000000 |
| 0111100110101101 | 0000000000000000001111111010011 1111111111111111101011111000000  1111111111111111111011110010011 | 0000000000000000001010000100000 |
| 1111001101011010 | 1111111111111111111011110010011 0000000000000000001010000100000  0000000000000000000101110110011 | 0000000000000000000101000010000 |

**2.2.5 Функціональна схема з відображенням управляючих сигналів**

***X1***

***X2***

***0***

***W2***

***R***

***ShL***

***n***

***RG3***

***2n***

***RG2***

***0***

MX

***0***

***W1***

***ShR***

***DR=’0’***

***RG1***

=1

***SM***

***0***

***2n***

***SM***

***0***

***2n***

***DL=p***

***2n***

***Y***

***X***

***V***

***S***

***2n+1***

***2n+1***

***2n+1***

*Рисунок 2.2.5-Функціональна схема*

**2.2.6 Закодований мікроалгоритм**

*Таблиця 2.2.6- Таблиця кодування мікрооперацій*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| RG3:=0;  RG1:=Y;  RG2:=X  RG2:=RG2+RG1  RG1:=0.r(RG1)  RG3:=l(RG3).SM(p)  RG2:=RG2++1 | R  W1  W2 ,V  W2  ShR  ShL  W2, S |  | RG2[2n+1]  RG3[n] | X1  X2 |
|  |

Початок

Z1

Z2

R,W1,W2, V

1

X1

Z4

Z3

W2,ShR,ShL

W2,S,ShR,ShL

X2

1

Z5

Кінець

*Рисунок 2.2.6- Закодований мікроалгоритм*

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

000

Z1

-

Z4

W2,ShR,ShL

Z3

W2,ShR,ShL, S

Z2

R,W1,W2, V

-

110

Q3Q2Q1

010

011

001

X2

X2

Z5

**-**

*Рисунок 2.2.7- Граф автомата Мура*

**2.2.8 Обробка порядків і нормалізація**

.

Нормалізація мантиси не потрібна.

MZ= , 111001101011010

Знак мантиси: .

**2.2.9 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 | 1 | 0 | 1 | 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 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.

,

.

X2 = 10010001,0001101

Y2 = 10100,0010000001

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 |
| 0.101000010000001 | 11 | П.С. |
| 0.010100001000000 | 10 | 🡪 |
| 0.001010000100000 | 01 | 🡪 |
| 0.000101000010000 | 00 | 🡪 |

1. Додавання мантис у модифікованому ДК.

Таблиця 3.2-Додавання мантис

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

1. Нормалізація результату (В ПК).

**3.2 Операційна схема**

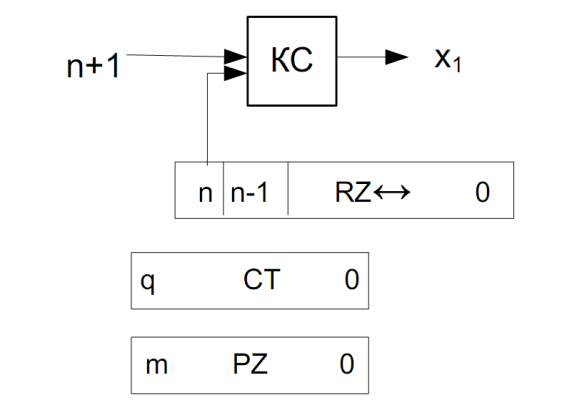
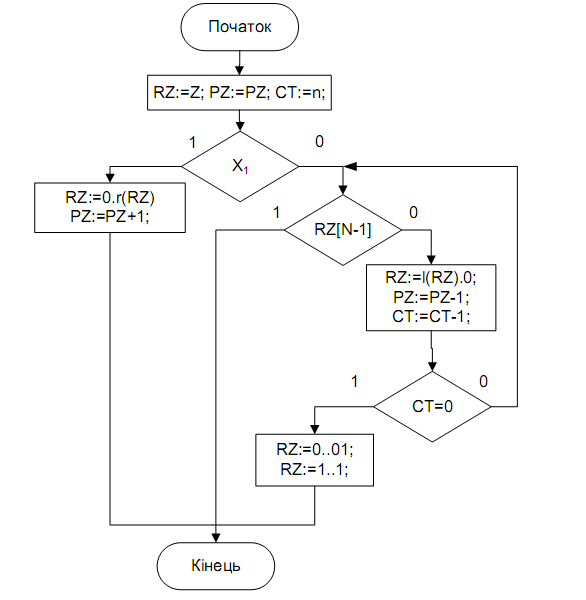
****

Рисунок 3.1-Операційна схема.

**3.3 Змістовний мікроалгоритм**

****

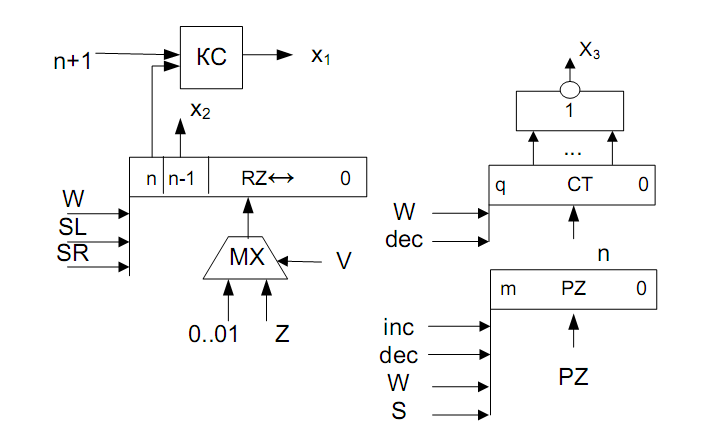
*Рисунок 3.2-Змістовний мікроалгоритм*

**3.4 Таблиця станів регістрів**

*Таблиця 3.4- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № | RZ | PZ | CT |
| ПС |  | 00001000 | 1111 |

**3.5 Функціональна схема з відображенням управляючих сигналів**

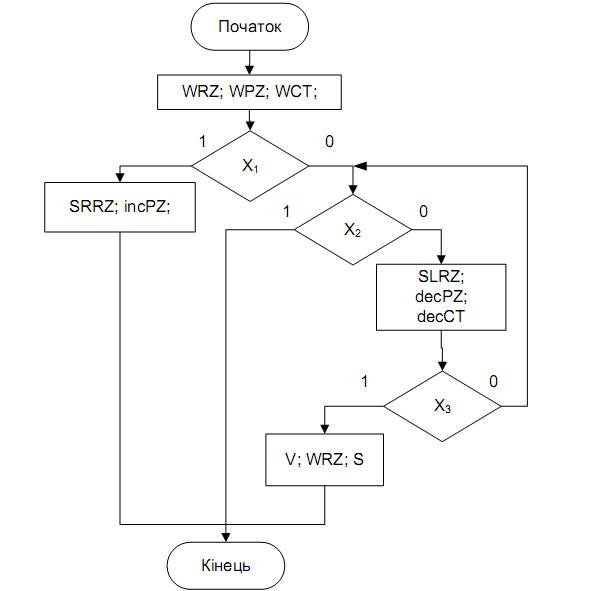
****

*Рисунок 3.3-Функціональна схема*

**3.6 Закодований мікроалгоритм**

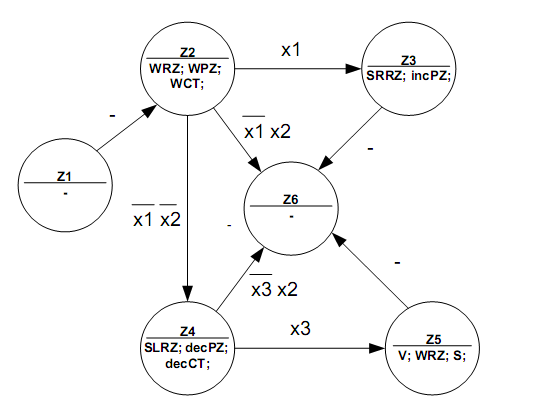
*Таблиця 3.5- Таблиця кодування мікрооперацій*

|  |
| --- |
|  |

****

*Рисунок 3.4- Закодований мікроалгоритм*

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

****

*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 Зн.М M

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

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

* 1. **Операційна схема**

***CT***

***0***

***SM***

***0***

***q***

***n+1***

***0***

***n+1***

2

n

11

***RZ***

1

***0***

2

2

***RX***

***n-1***

***0***

***RR***

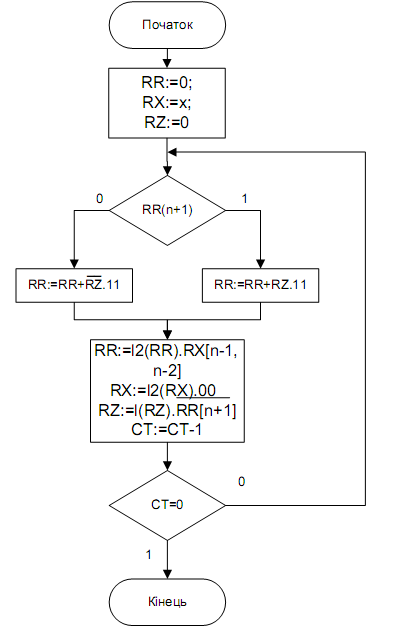
***n+1***

***0***

***n-1***

*Рисунок 4.2-Операційна схема*

**4.3 Змістовний мікроалгоритм**

****

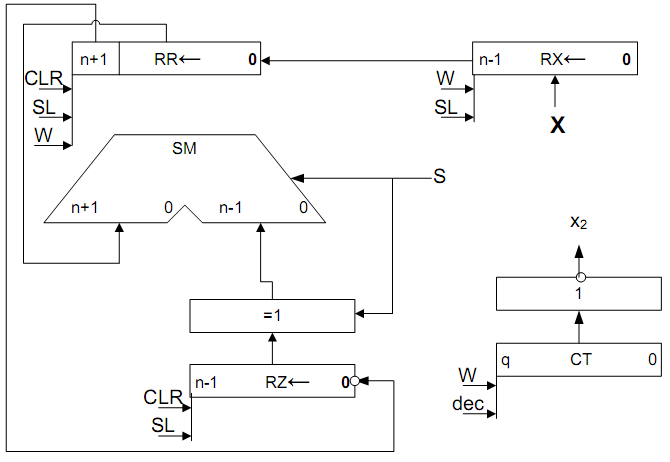
*Рисунок 4.3-Змістовний мікроалгоритм*

**4.4 Таблиця станів регістрів**

*Таблиця 4.4- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| RZ | RR | RX | CT |
| 0000000000000000 | 000000000000000000 | 0100100011000110 | 16 |
| 0000000000000000 | 000000000000000000 111111111111111111  111111111111111101 | 0010001100011000 | 15 |
| 0000000000000001 | 111111111111111101 000000000000000011  000000000000000000 | 1000110001100000 | 14 |
| 0000000000000010 | 000000000000000000 111111111111111011  111111111111101110 | 0011000110000000 | 13 |
| 0000000000000100 | 111111111111101110 000000000000001011  111111111111100100 | 1100011000000000 | 12 |
| 0000000000001000 | 111111111111100100 000000000000010011  111111111111011111 | 0001100000000000 | 11 |
| 0000000000010001 | 111111111111011111 000000000000100011  000000000000001000 | 0110000000000000 | 10 |
| 0000000000100010 | 000000000000001000 111111111110111011  111111111100001101 | 1000000000000000 | 9 |
| 0000000001000100 | 111111111100001101 000000000010001011  111111111001100010 | 0000000000000000 | 8 |
| 0000000010001000 | 111111111001100010 000000000100010011  111111110111010100 | 0000000000000000 | 7 |
| 0000000100010000 | 111111110111010100 000000001000100011  111111111111011100 | 0000000000000000 | 6 |
| 0000001000100001 | 111111111111011100 000000010001000011  000001000001111100 | 0000000000000000 | 5 |
| 0000010001000011 | 000001000001111100 111111011101111011  000001111111011100 | 0000000000000000 | 4 |
| 0000100010000111 | 000001111111011100 111110111011110011  000011101100111100 | 0000000000000000 | 3 |
| 0001000100001111 | 000011101100111100 111101110111100011  000110010001111100 | 0000000000000000 | 2 |
| 0010001000011111 | 000110010001111100 111011101111000011  001000000011111100 | 0000000000000000 | 1 |
| 0100010000111110 | 001000000011111100 110111011110000011  111110000111111100 | 0000000000000000 | 0 |

**4.5 Функціональна схема з відображенням управляючих сигналів**

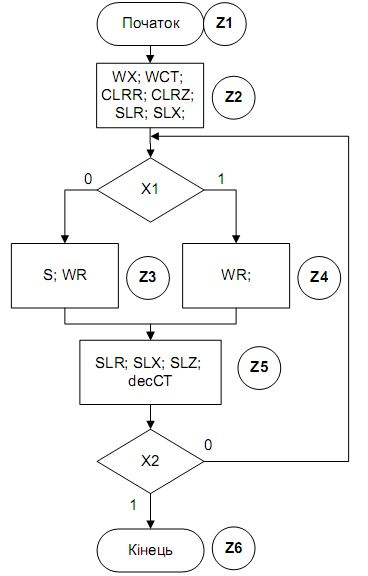
**

*Рисунок 4.5-Функціональна схема*

**4.6 Закодований мікроалгоритм**

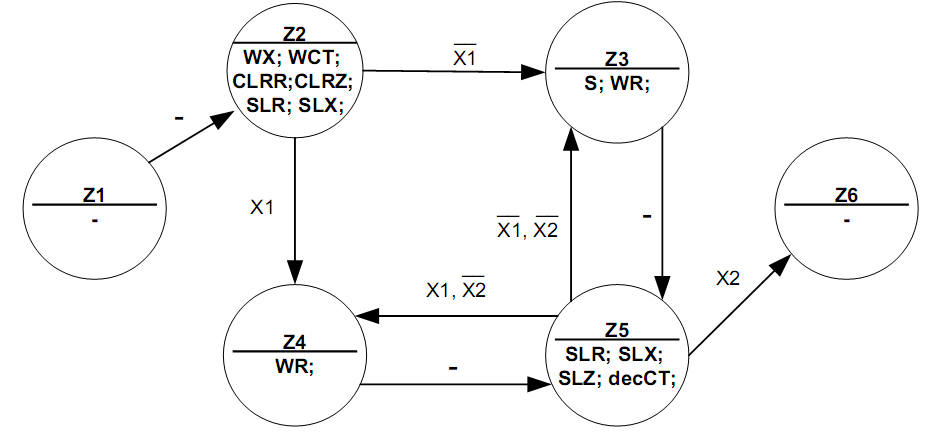
*Таблиця 4.2- Таблиця кодування мікрооперацій*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| RX:=X;  RR:=0;  RZ:=0  CT:=n  RR:=l2[RR].RX(n).RX(n-1)  RX:=l2[RX].0.0  R2:=R2++11  R2:=R2+R1+11  CT:=CT-1  RZ:=l[RZ]. | WX  CLR1  CLR2  WCT  SLR  SLX  S, WR  WR  Dec  SLZ |  | RR(n+1)  CT=0 | X1  X2 |

****

*Рисунок 4.6- Закодований мікроалгоритм*

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

****

*Рисунок 4.7- Граф автомата Мура*

**4.8 Обробка порядків**

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

Нормалізація: P = P-1 = 112

**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 | 1 | 1 | 1 | 1 | 0 |

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

x3x2x1 +1=101=6 – операція ділення 2 способом.

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

За закодованим мікроалгоритмом (Рис. 1.14) складемо таблицю:

*Таблиця 5.1 Таблиця кодування сигналів*

**5.2 Мікроалгоритм в термінах управляючого автомата**

|  |  |
| --- | --- |
| Виходи операційного автомата | Входи управляючого автомата |
| RG3[n] | X1 |
| RG2[2n] | X2 |

|  |  |
| --- | --- |
| Кодування мікрооперацій | |
| МО | Позначення |
| W2 | Y1 |
| R, W1, V | Y2 |
| ShL, ShR | Y3 |
| S | Y4 |

Зробимо автомат Мура циклічним задля зменшення кількості вершин.

Початок

Z1

Y1, Y2

Z2

1

X1

Y1, Y3

Y1, Y3, Y4

Z4

Z3

X2

1

Кінець

Z1

*Рисунок 5.1- Закодований мікроалгоритм*

Будуємо граф автомата Мура

Z2

Y1, Y2

00

01

Q2Q1

-

Z1

-

X2

X1

X2

Z3

Y1, Y3, Y4

Z4

Y1, Y3

11

10

*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 D-тригера.

*Таблиця 5.2-Структурна таблиця автомата*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Перехід | Q2Q1 | Q2Q1 | x1x2 | y1y2y3y4 | D2 | D1 |
| *z1z2* | 00 | 01 | -- | 0000 | 0 | 1 |
| *z2 z3* | 01 | 11 | 0- | 1100 | 1 | 1 |
| *z2 z4* | 01 | 10 | 1- | 1100 | 1 | 0 |
| *z3 z3* | 11 | 11 | 00 | 1010 | 1 | 1 |
| *z4 z4* | 10 | 10 | 10 | 1011 | 1 | 0 |
| *Z3 z4* | 11 | 10 | 10 | 1010 | 1 | 0 |
| *z4 z3* | 10 | 11 | 00 | 1011 | 1 | 1 |
| *z4 🡪 z1* | 10 | 00 | -1 | 1011 | 0 | 0 |
| *Z3 🡪 z1* | 11 | 00 | -1 | 1010 | 0 | 0 |

D-тригер:

**5.4 Синтез функцій виходів і переходів**

1

0

0

1

0

0

-

0

0

0

0

1

1

1

1

1

1

Q2

Q1

X2

X1

D­1

1

0

0

1

1

0

0

1

1

1

0

0

1

1

0

0

Q2

Q1

X2

X1

D2

*Рисунок 5.3- Діаграми Вейча*

D2=

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

**підключені до входів операційного автомата)**

***X1***

***X2***

***Y3***

***Y2***

***Y3***

***RG3***

***RG2***

***2n+1***

***0***

***n***

***2n***

***0***

***Y2***

MX

***X***

***Y4***

***DL=p***

***SM***

***SM***

***2n***

***2n***

***0***

***0***

***2n+1***

***2n+1***

=1

***Y2***

***Y3***

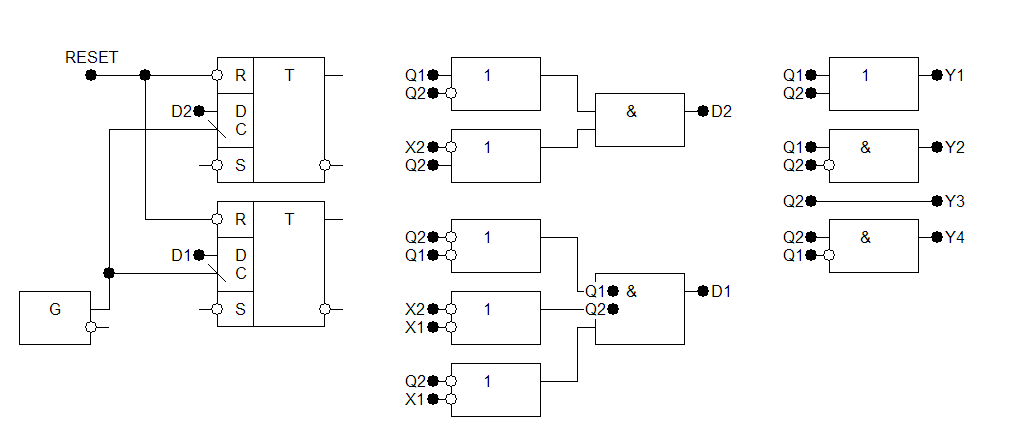
***DR=’0’***

***2n***

***0***

***RG1***

***Y***



*Рисунок 5.5- Функціональна схема пристрою*