НТУУ “КПІ”

Кафедра Обчислювальної техніки

# Розрахункова робота по курсу „КЛ-2”

Виконав

ст. 1 курсу

ФІОТ, гр. ІО — 82

Куцовола Віктора

Київ 2009

ЗМІСТ

[Завдання та обґрунтування варіанту **Ошибка! Закладка не определена.**](#_Toc231184700)

[1 Операція множення **Ошибка! Закладка не определена.**](#_Toc231184701)

[1.1 Перший спосіб множення **Ошибка! Закладка не определена.**](#_Toc231184702)

[1.1.1 Теоретичне обґрунтування способу **Ошибка! Закладка не определена.**](#_Toc231184703)

[1.1.2 Операційна схема **Ошибка! Закладка не определена.**](#_Toc231184704)

[1.1.3 Змістовний мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184705)

[1.1.4 Таблиця станів регістрів **Ошибка! Закладка не определена.**](#_Toc231184706)

[1.1.5 Функціональна схема пристрою **Ошибка! Закладка не определена.**](#_Toc231184707)

[1.1.6 Закодований мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184708)

[1.1.7 Граф управляючого автомата Мура **Ошибка! Закладка не определена.**](#_Toc231184709)

[1.1.8 Обробка порядків **Ошибка! Закладка не определена.**](#_Toc231184710)

[1.1.9 Форма подання результату в пам’ятті **Ошибка! Закладка не определена.**](#_Toc231184711)

[1.2 Другий спосіб множення **Ошибка! Закладка не определена.**](#_Toc231184712)

[1.2.1 Теоретичне обґрунтування способу **Ошибка! Закладка не определена.**](#_Toc231184713)

[1.2.2 Операційна схема **Ошибка! Закладка не определена.**](#_Toc231184714)

[1.2.3 Змістовний мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184715)

[1.2.4 Таблиця станів регістрів **Ошибка! Закладка не определена.**](#_Toc231184716)

[1.2.5 Функціональна схема пристрою **Ошибка! Закладка не определена.**](#_Toc231184723)

[1.2.6 Закодований мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184724)

[1.2.7 Граф управляючого автомата Мура **Ошибка! Закладка не определена.**](#_Toc231184725)

[1.2.8 Обробка порядків **Ошибка! Закладка не определена.**](#_Toc231184726)

[1.2.9 Форма подання результату в пам’ятті **Ошибка! Закладка не определена.**](#_Toc231184727)

[1.3 Третій спосіб множення **Ошибка! Закладка не определена.**](#_Toc231184728)

[1.3.1 Теоретичне обґрунтування способу **Ошибка! Закладка не определена.**](#_Toc231184729)

[1.3.2 Операційна схема **Ошибка! Закладка не определена.**](#_Toc231184730)

[1.3.3 Змістовний мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184731)

[1.3.4 Таблиця станів регістрів **Ошибка! Закладка не определена.**](#_Toc231184732)

[1.4.5 Функціональна схема пристрою **Ошибка! Закладка не определена.**](#_Toc231184738)

[1.4.6 Закодований мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184739)

[1.4.7 Граф управляючого автомата Мура **Ошибка! Закладка не определена.**](#_Toc231184740)

[1.4.8 Обробка порядків **Ошибка! Закладка не определена.**](#_Toc231184741)

[1.4.9 Форма подання результату в пам’ятті **Ошибка! Закладка не определена.**](#_Toc231184742)

[1.4 Четвертий спосіб множення **Ошибка! Закладка не определена.**](#_Toc231184743)

[1.4.1 Теоретичне обґрунтування способу **Ошибка! Закладка не определена.**](#_Toc231184744)

[1.4.2 Операційна схема **Ошибка! Закладка не определена.**](#_Toc231184745)

[1.4.3 Змістовний мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184746)

[1.4.4 Таблиця станів регістрів **Ошибка! Закладка не определена.**](#_Toc231184747)

[1.4.5 Функціональна схема пристрою **Ошибка! Закладка не определена.**](#_Toc231184754)

[1.4.6 Закодований мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184755)

[1.4.7 Граф управляючого автомата Мура **Ошибка! Закладка не определена.**](#_Toc231184756)

[1.4.8 Обробка порядків **Ошибка! Закладка не определена.**](#_Toc231184757)

[1.4.9 Форма подання результату в пам’ятті **Ошибка! Закладка не определена.**](#_Toc231184758)

[2 Операція ділення **Ошибка! Закладка не определена.**](#_Toc231184759)

[2.1 Перший спосіб ділення **Ошибка! Закладка не определена.**](#_Toc231184760)

[2.1.1 Теоретичне обґрунтування способу **Ошибка! Закладка не определена.**](#_Toc231184761)

[2.1.2 Операційна схема **Ошибка! Закладка не определена.**](#_Toc231184762)

[2.1.3 Змістовний мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184763)

[2.1.4 Таблиця станів регістрів **Ошибка! Закладка не определена.**](#_Toc231184764)

[2.1.5 Функціональна схема пристрою **Ошибка! Закладка не определена.**](#_Toc231184765)

[2.1.6 Закодований мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184766)

[2.1.7 Граф управляючого автомата Мура **Ошибка! Закладка не определена.**](#_Toc231184767)

[2.1.8 Обробка порядків **Ошибка! Закладка не определена.**](#_Toc231184768)

[2.1.9 Форма подання результату в пам’ятті **Ошибка! Закладка не определена.**](#_Toc231184769)

[2.2 Другий спосіб ділення **Ошибка! Закладка не определена.**](#_Toc231184770)

[2.2.1 Теоретичне обґрунтування способу **Ошибка! Закладка не определена.**](#_Toc231184771)

[2.2.2 Операційна схема **Ошибка! Закладка не определена.**](#_Toc231184772)

[2.2.3 Змістовний мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184773)

[2.2.4 Таблиця станів регістрів **Ошибка! Закладка не определена.**](#_Toc231184774)

[2.2.5 Функціональна схема пристрою **Ошибка! Закладка не определена.**](#_Toc231184775)

[2.2.6 Закодований мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184776)

[2.2.7 Граф управляючого автомата Мура **Ошибка! Закладка не определена.**](#_Toc231184777)

[2.2.8 Обробка порядків **Ошибка! Закладка не определена.**](#_Toc231184778)

[2.2.9 Форма подання результату в пам’ятті **Ошибка! Закладка не определена.**](#_Toc231184779)

[3 Операція добування кореня 2](#_Toc231184780)

[3.1 Теоретичне обґрунтування способу 2](#_Toc231184781)

[3.2 Операційна схема 2](#_Toc231184782)

[3.3 Змістовний мікроалгоритм 2](#_Toc231184783)

[3.4 Таблиця станів регістрів 2](#_Toc231184784)

[3.5 Функціональна схема пристрою 2](#_Toc231184785)

[3.6 Закодований мікроалгоритм 2](#_Toc231184786)

[3.7 Граф управляючого автомата Мура 2](#_Toc231184787)

[3.8 Обробка порядків 2](#_Toc231184788)

[3.9 Форма подання результату в пам’ятті 2](#_Toc231184789)

[4 Операція додавання **Ошибка! Закладка не определена.**](#_Toc231184790)

[4.1 Теоретичне обґрунтування способу **Ошибка! Закладка не определена.**](#_Toc231184791)

[4.2 Операційна схема **Ошибка! Закладка не определена.**](#_Toc231184792)

[4.3 Змістовний мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184793)

[4.4 Таблиця станів регістрів **Ошибка! Закладка не определена.**](#_Toc231184794)

[4.5 Функціональна схема пристрою **Ошибка! Закладка не определена.**](#_Toc231184800)

[4.6 Закодований мікроалгоритм **Ошибка! Закладка не определена.**](#_Toc231184801)

[4.7 Граф управляючого автомата Мура **Ошибка! Закладка не определена.**](#_Toc231184802)

[4.8 Обробка порядків **Ошибка! Закладка не определена.**](#_Toc231184803)

[4.9 Форма подання результату в пам’ятті **Ошибка! Закладка не определена.**](#_Toc231184804)

[5 Синтез управляючого автомата для операційного пристрою нормалізації результату додавання 2](#_Toc231184805)

[5.1 Таблиця співвідношення управляючих входів операційного автомата і виходів управляючого автомата 2](#_Toc231184806)

[5.2 Мікроалгоритм в термінах управляючого автомата 2](#_Toc231184807)

[5.3 Структурна таблиця автомата 2](#_Toc231184808)

[5.4 Синтез функцій виходів і переходів 2](#_Toc231184809)

[5.5 Функціональна схема пристою (виходи управляючого автомата підключені до входів операційного автомата) 2](#_Toc231184810)

[Висновок 2](#_Toc231184811)

[Список літератури: 2](#_Toc231184812)

Варіант завдання

Перевести номер залікової книжки в двійкову систему. Записати два двійкових числа:

 і ,

де  - двійкові цифри номера залікової книжки у двійковій системі числення ( - молодший розряд).

Завдання

1. Числа  і  в прямому коді записати у формі з плаваючою комою (з порядком і мантисою, а також з характеристикою та мантисою), як вони зберігаються у пам’яті. На порядок (характеристику) відвести 8 розрядів, на мантису 16 розрядів (з урахуванням знакових розрядів). (0,5)

2. Виконати 8 операцій з числами  і  з плаваючою комою (чотири способи множення, два способи ділення, додавання та добування кореня з ). Номери операцій (для п.3) відповідають порядку переліку (наприклад, 1 – множення першим способом; 6 – ділення другим способом; 8 – добування кореня). Для обробки мантис кожної операції, подати:

2.1 теоретичне обґрунтування способу; (0,2)

2.1 операційну схему; (0,2)

2.2 змістовний мікроалгоритм; (0,2)

2.3 таблицю станів регістрів (лічильника), довжина яких забезпечує одержання 15 основних розрядів мантиси результату; (1,5)

2.4 функціональну схему з відображенням управляючих сигналів; (0.5)

2.5 закодований мікроалгоритм (мікрооперації замінюються управл. сигналами); (0.3)

2.6 граф управляючого автомата Мура з кодами вершин; (0,5)

2.7 обробку порядків (показати у довільній формі); (0,5)

2.8 форму запису нормалізованого результату з плаваючою комою в пам’ять. (0,1)

Операцію додавання до етапу нормалізації результату можна проілюструвати у довільній формі. Вказані пункти виконати для етапу нормалізації результату з урахуванням можливого нулевого результату.

3. Для операції з двійковим номером +1 побудувати управляючий автомат Мура на тригерах ( =00 відповідає RS-тригеру; 01 – D-тригеру; 10 – JK-тригеру; 11 – T-тригеру) і елементах булевого базису. (1,5)

Обгрунтування варіанту:

Номер залікової книжки: 82132=2

*X2*= -10010001,1001101

*Y2*=+10100,0011001011

*X2*пк=1.10010001,1001101

*Y2*пк=0.10100,0011001011

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

*P*=810=10002 *M*=1.100100011001101

|  |  |
| --- | --- |
| 0 | 0001000 |

|  |  |
| --- | --- |
| 1 | 100100011001101 |

*X2* =

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

*Р*=510=1012 *М*=0.101000011001011

|  |  |
| --- | --- |
| 0 | 0000101 |

|  |  |
| --- | --- |
| 0 | 101000011001011 |

*Y2*=

X2 у формі з плаваючою комою (з характеристикою та мантисою):

*С*=*P*-2n-1=810-12810=-12010=[1.100001112]ДК

|  |  |
| --- | --- |
| 1 | 1000011 |

|  |  |
| --- | --- |
| 1 | 010000110010110 |

*X2*=

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

*С*=*P*-2n-1=510-12810=-12310=[1.100001002]ДК

|  |  |
| --- | --- |
| 1 | 1000011 |

|  |  |
| --- | --- |
| 0 | 010000110010110 |

*Y2*=

1. Операція множення

Під час множення чисел у прямих кодах знакові і основні розряди обробляються окремо. Для визначення знака добутку здійснюється підсумування по модулю 2 цифр, що розміщуються в знакових розрядах співмножників.

Будемо вважати, що множене *Y* та множник *Х* – правильні двійкові дроби вигляду *Х = 0,x1, x2,…, xn Y=0, y1, y2,…, yn*, де *xi, yi*  {0,1}.

*Z=YX=Yxn*2-n+ *Yxn-1*2-n+1+…+ *Yxi*2-i+…+ *Yx1*2-1 (Б-1)

Множення чисел *Y* та *X* може бути реалізоване шляхом виконання визначеного циклічного процесу, характер якого залежить від конкретної форми виразу. Один цикл множення складається з додавання чергового часткового добутку, що являє собою добуток множеного на одну цифру множника, до суми часткових добутків.

* 1. Перший спосіб множення
     1. Теоретичне обґрунтування способу

Вираз (Б-1) можна подати у вигляді:

*Z=YX=*((*…*((0*+Yxn*))2-1+*Yxn-1*)2-1*+*…+*Yxi*)2-1+…*Yx1*)2-1,

з отриманого виразу виходить, що отримання суми часткових добутків в *і*=1,n , зводиться до обчислення виразу

*Zi=*(*Zi-1+Yxn-i+1*)2-1

з початковими значеннями *i*=1, *Z0=*0, при чому *Zn=Z=YX.*

У розглянутому способі множення здійснюється з молодших розрядів множника, сума часткових добутків зсувається вправо, а множник залишається нерухомим.

* + 1. Операційна схема



* + 1. Змістовний мікроалгоритм



*Рис. 1.2. Змістовний мікроалгоритм*

* + 1. **Таблиця станів регістрів (Таблиця 1.1)**

*Таблиця 1.1. Таблиця станів регістрів*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № циклу | RG1 | RG2 | RG3 | CT |
| ПС | 0000000000000000 | 10010001100110**1** | 101000011001011 | 1111 |
| 1 |  | 11001000110011**0** |  | 1110 |
| 2 | 0001010000110010 | 11100100011001**1** |  | 1101 |
| 3 | 0110010011111101  0011001001111110 | 11110010001100**1** |  | 1100 |
| 4 | 1000001101001001  0100000110100100 | 11111001000110**0** |  | 1011 |
| 5 | 0010000011010010 | 01111100100011**0** |  | 1010 |
| 6 | 0001000001101001 | 00111110010001**1** |  | 1001 |
| 7 | 0110000100110100  0011000010011010 | 00011111001000**1** |  | 1000 |
| 8 | 1000000101100101  0100000010110010 | 10001111100100**0** |  | 0111 |
| 9 | 0010000001011001 | 01000111110010**0** |  | 0110 |
| 10 | 0001000000101100 | 10100011111001**0** |  | 0101 |
| 11 | 0000100000010110 | 01010001111100**1** |  | 0100 |
| 12 | 0101100011100001  0010110001110000 | 10101000111110**0** |  | 0011 |
| 13 | 0001011000111000 | 01010100011111**0** |  | 0010 |
| 14 | 0000101100011100 | 00101010001111**1** |  | 0001 |
| 15 | 0101101111100111  **0,010110111110011** | **100101010001111** |  | 0000 |

* + 1. **Функціональна схема пристрою**



* + 1. Закодований мікроалгоритм



* + 1. Граф управляючого автомата Мура



* + 1. **Обробка порядків**

Порядок результату:  
*Рz=Px+Py=*8+5=1310=11012

Мантиса одержаного числа:

*М*=0,010110111110011100101010001111 – мантиса не є нормалізованою, зсуваємо її на один розряд ліворуч і робимо корекцію порядку *Pz-*1=1210=11002. Отримаємо:

*Z*= 0 00001100 1 1011011111001110

* 1. Другий спосіб множення
     1. Теоретичне обґрунтування способу

Вираз (Б-1) можна подати у вигляді:

*Z=*((*…*((0*+*Y2-nxn))+Y2-n+1xn*-1*)*+*…+*Y*2-1 *x1*.

Очевидно, що процес множення може бути зведений до n-кратного виконання циклу

*Zi=Zi-1+Yi xn-i+1­*, *Yi=*2*Yi-1*.

З початковими значеннями *і=*1, *Y0*=*Y*2-n, *Z0=*0.

У розглянутому способі множення здійснюється з молодших розрядів, множене зсувається вліво, а сума часткових добутків залишається нерухомою.

* + 1. Операційна схема



*Рис.1.6. Оперційна схема*

* + 1. Змістовний мікроалгоритм



*Рис. 1.7. Змістовний мікроалгоритм*

* + 1. Таблиця станів регістрів

|  |  |  |  |
| --- | --- | --- | --- |
| № | RG1 | RG3 | RG2 |
| ПС |  | 000000000000000101000011001011 | 10010001100110**1** |
| 1 |  | 000000000000001010000110010110 | 01001000110011**0** |
| 2 |  | 000000000000010100001100101100 | 00100100011001**1** |
| 3 |  | 000000000000101000011001011000 | 00010010001100**1** |
| 4 |  | 000000000001010000110010110000 | 00001001000110**0** |
| 5 |  | 000000000010100001100101100000 | 00000100100011**0** |
| 6 |  | 000000000101000011001011000000 | 00000010010001**1** |
| 7 |  | 000000001010000110010110000000 | 00000001001000**1** |
| 8 |  | 000000010100001100101100000000 | 00000000100100**0** |
| 9 |  | 000000101000011001011000000000 | 00000000010010**0** |
| 10 |  | 000001010000110010110000000000 | 00000000001001**0** |
| 11 |  | 000010100001100101100000000000 | 00000000000100**1** |
| 12 |  | 000101000011001011000000000000 | 00000000000010**0** |
| 13 |  | 001010000110010110000000000000 | 00000000000001**0** |
| 14 |  | 010100001100101100000000000000 | 00000000000000**1** |
| 15 |  | 101000011001011000000000000000 | 000000000000000 |

* + 1. Функціональна схема пристрою



* + 1. Закодований мікроалгоритм



* + 1. Граф управляючого автомата Мура



* + 1. Обробка порядків

Порядок результату:  
*Рz=Px+Py=*8+5=1310=11012

Мантиса одержаного числа:

*М*=0,010110111110011100101010001111 – мантиса не є нормалізованою, зсуваємо її на один розряд ліворуч і робимо корекцію порядку *Pz-*1=1210=11002. Отримаємо:

*Z*= 0 00001100 1 1011011111001110

* 1. Третій спосіб множення
     1. Теоретичне обґрунтування способу

Вираз (Б-1) можна подати у вигляді:

*Z=*((*…*((0*+*Y2-nx1))2+Y2-nx2)2*+*…+*Y*2-n *xi*)2+*Y*2-n*xn*.

Сума часткових добутків у *і*-му циклі *i=*1*,n* можна одержати за виразом

*Zi=*2*Zi-1+Y*2-n*xi*,

з початковими значеннями *і=*1, *Z0=*0.

У розглянутому способі множення здійснюється зі старших розрядів множника, сума часткових добутків зсувається вліво, а множене нерухоме.

* + 1. Операційна схема



* + 1. Змістовний мікроалгоритм



* + 1. Таблиця станів регістрів

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | RG1 | RG3 | RG2 | CT |
| ПС | 000000000000000000000000000000 | 101000011001011 | **1**00100011001101 | 1111 |
| 1 |  | 101000011001011 | **0**01000110011010 | 1110 |
| 2 |  | 101000011001011 | **0**10001100110100 | 1101 |
| 3 |  | 101000011001011 | **1**00011001101000 | 1100 |
| 4 |  | 101000011001011 | **0**00110011010000 | 1011 |
| 5 |  | 101000011001011 | **0**01100110100000 | 1010 |
| 6 |  | 101000011001011 | **0**11001101000000 | 1001 |
| 7 |  | 101000011001011 | **1**10011010000000 | 1000 |
| 8 |  | 101000011001011 | **1**00110100000000 | 0111 |
| 9 |  | 101000011001011 | **0**01101000000000 | 0110 |
| 10 |  | 101000011001011 | **0**11010000000000 | 0101 |
| 11 |  | 101000011001011 | **1**10100000000000 | 0100 |
| 12 |  | 101000011001011 | **1**01000000000000 | 0011 |
| 13 |  | 101000011001011 | **0**10000000000000 | 0010 |
| 14 |  | 101000011001011 | **1**00000000000000 | 0001 |
| 15 |  | 101000011001011 | 000000000000000 | 0000 |

* + 1. Функціональна схема пристрою



* + 1. Закодований мікроалгоритм



* + 1. Граф управляючого автомата Мура



* + 1. Обробка порядків

Порядок результату:  
*Рz=Px+Py=*8+5=1310=11012

Мантиса одержаного числа:

*М*=0,010110111110011100101010001111 – мантиса не є нормалізованою, зсуваємо її на один розряд ліворуч і робимо корекцію порядку *Pz-*1=1210=11002. Отримаємо:

*Z*= 0 00001100 1 1011011111001110

* 1. Четвертий спосіб множення
     1. Теоретичне обґрунтування способу

Вираз (Б-1) можна подати у вигляді:

*Z=*((*…*((0*+*Y2-nx1)+Y2-2x2)*+*…+*Y*2-i *xi*)+*Y*2-n*xn*.

У цьому випадку процес множення може бути зведений до n-кратного виконання циклу

*Zi=Zi-1+Yi-*1*xi*, *Yi=Yi-*12-1

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

У розглянутому способі множення здійснюється зі старших розрядів множника, сума часткових добутків залишається нерухомою, а множене зсувається вправо.

* + 1. Операційна схема



* + 1. Змістовний мікроалгоритм



* + 1. Таблиця станів регістрів

|  |  |  |  |
| --- | --- | --- | --- |
| № | RG1 | RG3 | RG2 |
| ПС |  | 010100001100101100000000000000 | **1**00100011001101 |
| 1 |  | 001010000110010110000000000000 | **0**01000110011010 |
| 2 | 010100001100101100000000000000 | 000101000011001011000000000000 | **0**10001100110100 |
| 3 | 010100001100101100000000000000 | 000010100001100101100000000000 | **1**00011001101000 |
| 4 |  | 000001010000110010110000000000 | **0**00110011010000 |
| 5 |  | 000000101000011001011000000000 | **0**01100110100000 |
| 6 |  | 000000010100001100101100000000 | **0**11001101000000 |
| 7 |  | 000000001010000110010110000000 | **1**10011010000000 |
| 8 |  | 000000000101000011001011000000 | **1**00110100000000 |
| 9 | + | 000000000010100001100101100000 | **0**01101000000000 |
| 10 |  | 000000000001010000110010110000 | **0**11010000000000 |
| 11 |  | 000000000000101000011001011000 | **1**10100000000000 |
| 12 |  | 000000000000010100001100101100 | **1**01000000000000 |
| 13 |  | 000000000000001010000110010110 | **0**10000000000000 |
| 14 |  | 000000000000000101000011001011 | **1**00000000000000 |
| 15 |  | 000000000000000010100001100101 | **0**00000000000000 |

* + 1. Функціональна схема пристрою



* + 1. Закодований мікроалгоритм



* + 1. Граф управляючого автомата Мура



* + 1. Обробка порядків

Порядок результату:  
*Рz=Px+Py=*8+5=1310=11012

Мантиса одержаного числа:

*М*=0,010110111110011100101010001111 – мантиса не є нормалізованою, зсуваємо її на один розряд ліворуч і робимо корекцію порядку *Pz-*1=1210=11002. Отримаємо:

*Z*= 0 00001100 1 1011011111001110

1. Операція ділення
   1. Перший спосіб ділення
      1. Теоретичне обґрунтування способу

Під час реалізації ділення за першим способом здійснюється зсув вліво залишку при нерухомому дільнику, такий спосіб називається *діленням зі зсувом залишку*. На операційній схемі показаний принцип побудови пристрою.

* + 1. Операційна схема



* + 1. Змістовний мікроалгоритм



* + 1. Таблиця станів регістрів

|  |  |  |  |
| --- | --- | --- | --- |
| № циклу | RG1 | RG2 | RG3 |
| ПС | 0000000000000000 |  | 00101000011001011ПК  11010111100110101ДК |
| 1 | 0000000000000001 |  |  |
| 2 | 0000000000000011 |  |  |
| 3 | 0000000000000111 |  |  |
| 4 | 0000000000001111 |  |  |
| 5 | 0000000000011110 |  |  |
| 6 | 0000000000111100 |  |  |
| 7 | 0000000001111001 |  |  |
| 8 | 0000000011110011 |  |  |
| 9 | 0000000111100110 |  |  |
| 10 | 0000001111001101 |  |  |
| 11 | 0000011110011010 | 0 |  |
| 12 | 0000111100110101 |  |  |
| 13 | 0001111001101010 |  |  |
| 14 | 0011110011010101 |  |  |
| 15 | 0111100110101011 |  |  |
| 16 | 1111001101010110 |  |  |

* + 1. Функціональна схема пристрою



* + 1. Закодований мікроалгоритм



* + 1. Граф управляючого автомата Мура



* + 1. Обробка порядків

Порядок результату:  
*Рz=Px-Py=*8-5=310=112

Мантиса одержаного числа:

*М*=0,111001101010110

Маємо:

*Z*= 0 00000011 1 111001101010110

* 1. Другий спосіб ділення
     1. Теоретичне обґрунтування способу

Під час реалізації ділення другим способом, який називається діленням зі зсувом дільника, збільшується розрядність регістрів RG1, RG3 і суматора SM. Процеси додавання-віднімання і зсуву можуть бути сполученні в часі. Отже, для ділення за другим способом час одержання цифри результату дорівнює t=tД. Цифра результату формується на виході переносу суматора SM.

* + 1. Операційна схема



* + 1. Змістовний мікроалгоритм



* + 1. Таблиця станів регістрів

|  |  |  |  |
| --- | --- | --- | --- |
| № | RG1 | RG2 | RG3 |
| ПС | 000000000000000**1** |  | 001010000110010110000000000000110101111001101010000000000000 |
| 1 | 00000000000000**1**1 |  | 000101000011001011000000000000  111010111100110101000000000000 |
| 2 | 0000000000000**1**11 |  | 000010100001100101100000000000  111101011110011010100000000000 |
| 3 | 000000000000**1**111 |  | 000001010000110010110000000000  111110101111001101010000000000 |
| 4 | 00000000000**1**1110 |  | 000000101000011001011000000000  111111010111100110101000000000 |
| 5 | 0000000000**1**11100 |  | 000000010100001100101100000000  111111101011110011010100000000 |
| 6 | 000000000**1**111001 |  | 000000001010000110010110000000  111111110101111001101010000000 |
| 7 | 00000000**1**1110011 |  | 000000000101000011001011000000  111111111010111100110101000000 |
| 8 | 0000000**1**11100110 |  | 000000000010100001100101100000  111111111101011110011010100000 |
| 9 | 000000**1**111001101 |  | 000000000001010000110010110000  111111111110101111001101010000 |
| 10 | 00000**1**1110011010 |  | 000000000000101000011001011000  111111111111010111100110101000 |
| 11 | 0000**1**11100110101 |  | 000000000000010100001100101100  111111111111101011110011010100 |
| 12 | 000**1**111001101010 |  | 000000000000001010000110010110  111111111111110101111001101010 |
| 13 | 00**1**1110011010101 |  | 000000000000000101000011001011  111111111111111010111100110101 |
| 14 | 0**1**11100110101011 |  | 000000000000000010100001100101  111111111111111101011110011010 |
| 15 | **1**111001101010110 |  | 000000000000000001010000110010  111111111111111110101111001101 |

* + 1. Функціональна схема пристрою



* + 1. Закодований мікроалгоритм



* + 1. Граф управляючого автомата Мура



* + 1. Обробка порядків

Порядок результату:  
*Рz=Px-Py=*8-5=310=112

Мантиса одержаного числа:

*М*=0,111001101010110

Маємо:

*Z*= 0 00000011 1 111001101010110

1. Операція додавання
   1. Теоретичне обґрунтування способу

При додаванні спочатку потрібно вирівняти порядки доданків: менший порядок збільшується, а мантиса зсувається. Потім додаються мантиси, а порядки залишаються сталими, після чого виконується нормалізація результату,з урахуванням того,що мантиса може дорівнювати 0.

* 1. Операційна схема



Рисунок 4.1 – Операційна схема пристрою додавання

* 1. Змістовний мікроалгоритм



Рисунок 1.2 – Блок-схема алгоритму додавання

* 1. Операція додавання:

* 1. Функціональна схема пристрою



Рисунок 4.3 – Функціональна схема пристрою додавання

* 1. Закодований мікроалгоритм



Рисунок 4.4 – Закодований мікроалгоритм додавання

* 1. Граф управляючого автомата Мура



Рисунок 4.5 – Граф управляючого автомата для додавання

* 1. Обробка порядків

PX=00001000 MX=11.100100011001101

РY=00000101 МY=00. 101000011001011

PY’=00001000 MY’=00. 010100001100101

PZ=00001000 MZ=11.111000100110010

* 1. Форма подання результату в пам’ятті

Z = 0 00001000 1 111000100110010

# Операція добування кореня

## Теоретичне обґрунтування способу

Розглядаємо операцію , де X<1.

Результат формується порозрядно, тому ;

Алгоритм має наступні етапи:

Цей спосіб також не є регулярним, тому його модифікують так, щоб остачу не відновлювати.

## Операційна схема



Рисунок 3.1 – Операційна схема пристрою добування кореня

## Змістовний мікроалгоритм



Рисунок 3.2 – Блок-схема алгоритму добування кореня

## Таблиця станів регістрів

Таблиця 3.1 – Таблиця станів регістрів для добування кореня

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № циклу | RG1 | RG2 | RG3 | CT |
| ПС | 000000000000000 | 00000000000000000  00000000000000001 | 010100001100110  010000110011000 | 1111 |
| 1 | 000000000000001 |  | 000011001100000 | 1110 |
| 2 | 000000000000010 |  | 001100110000000 | 1101 |
| 3 | 000000000000100 |  | 110011000000000 | 1100 |
| 4 | 000000000001000 |  | 001100000000000 | 1011 |
| 5 | 000000000010001 |  | 110000000000000 | 1010 |
| 6 | 000000000100011 |  | 000000000000000 | 1001 |
| 7 | 000000001000111 |  | 000000000000000 | 1000 |
| 8 | 000000010001111 |  | 000000000000000 | 0111 |
| 9 | 000000100011111 |  | 000000000000000 | 0110 |
| 10 | 000001000111111 |  | 000000000000000 | 0101 |
| 11 | 000010001111111 |  | 000000000000000 | 0100 |
| 12 | 000100011111111 |  | 000000000000000 | 0011 |
| 13 | 001000111111111 |  | 000000000000000 | 0010 |
| 14 | 010001111111111 |  | 000000000000000 | 0001 |
| 15 | 100011111111111 |  | 000000000000000 | 0000 |

## Функціональна схема пристрою



Рисунок 3.3 – Функціональна схема пристрою добування кореня

## Закодований мікроалгоритм



Рисунок 3.4 – Закодований мікроалгоритм добування кореня

## Граф управляючого автомата Мура



Рисунок 3.5 – Граф управляючого автомата для добування кореня

## Обробка порядків

Порядок результату:  
Рz=Py/2=00001000/2=00000100

Мантиса одержаного числа:

М=0.100011111111111– мантиса завжди нормалізована. Отримаємо:

Z= 0.00000011 0.100011111111111

## Форма подання результату в пам’ятті

Z= 0.00000011 0.100011111111111

# Синтез управляючого автомата для операційного пристрою нормалізації результату додавання

## Таблиця співвідношення управляючих входів операційного автомата і виходів управляючого автомата

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

|  |  |
| --- | --- |
| управляючі входи | виходи управляючого автомата |
| WM,WP | Y1 |
| SHRM,incP | Y2 |
| WCT | Y3 |
| SHLM,incP,decCT | Y4 |

## Мікроалгоритм в термінах управляючого автомата



Рисунок 5.1 – Закодований мікроалгоритм додавання

## Структурна таблиця автомата

Таблиця 5.2 - Структурна таблиця автомата

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Q3Q2Q1 | Q3Q2Q1 | OV F N | Y1Y2Y3Y4 | I3K3 | I2K2 | I1K1 |
| 0 0 0 | 0 0 1 | - - - | 0 0 0 0 | 0 - | 0 - | 1 - |
| 0 0 1 | 0 1 1 | 1 - - | 1 0 0 0 | 0 - | 1 - | - 0 |
| 0 0 1 | 1 0 1 | 0 - - | 1 0 0 0 | 1 - | 0 - | - 0 |
| 0 1 1 | 1 1 1 | - - - | 0 1 0 0 | 1 - | - 0 | - 0 |
| 1 0 1 | 1 1 0 | - 0 0 | 0 0 1 0 | - 0 | 1 - | - 1 |
| 1 0 1 | 1 1 1 | - 1 1 | 0 0 1 0 | - 0 | 1 - | - 0 |
| 1 1 0 | 1 1 0 | - 0 0 | 0 0 0 1 | - 0 | - 0 | 0 - |
| 1 1 0 | 1 1 1 | - 1 1 | 0 0 0 1 | - 0 | - 0 | 1 - |

## Синтез функцій виходів і переходів

   

Рисунок 5.2 – Діаграми Вейча для функцій виходу

     

Рисунок 5.3 – Діаграми Вейча для функцій переходу

## Функціональна схема пристою (виходи управляючого автомата підключені до входів операційного автомата)





Рисунок 5.4 – Функціональна схема управляючого автомата

# Висновок

В даній роботі результати множення і ділення різними способами співпали, тому операції виконано правильно. Було удосконалено і систематизовано навички виконання арифметичних операцій в двійкових кодах над числами з плаваючою комою і побудови автоматів за графом.

# Список літератури:

1. Жабін В.І., Жуков І.А., Клименко І.А., Стіренко С.Г. – Прикладна теорія цифрових автоматів: навчальний посібник. – книжкове видавництво НАУ, 2007. – 364с.
2. Жабін В.І., Жуков І.А., Клименко І.А., Стіренко С.Г. – Арифметичні та управляючі пристрої цифрових ЕОМ: навчальний посібник. – К.:ВЕК+, 2008. – 176с.
3. Самофалов К.Г., Романкевич А.М., Валуйский В.Н., Каневский Ю.С., Пиневич М.М. Прикладная теория цифровых автоматов.– К.: Вища школа, 1987. – 375 с.