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

Петрук Вадим Олександрович

Керівник: Жабін В.І.

2010 рік

**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 відвести 8 розрядів, на мантису 16 розрядів (з урахуванням знакових розрядів).

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, 6 – ділення другим способом). Для обробки мантис кожної операції, подати:

2.1 теоретичне обґрунтування способу;

2.1 операційну схему;

2.2 змістовний мікроалгоритм;

2.3 таблицю станів регістрів (лічильника), довжина яких забезпечує одержання 15 основних

розрядів мантиси результату;

2.4 функціональну схему з відображенням управляючих сигналів;

2.5 закодований мікроалгоритм (мікрооперації замінюються управляючими сигналами);

2.6 граф управляючого автомата Мура з кодами вершин;

2.7 обробку порядків (показати у довільній формі);

2.8 форму запису нормалізованого результату з плаваючою комою в пам’ять.

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

3. Для операції з номером  побудувати управляючий автомат Мура на тригерах (тип вибрати самостійно) і елементах булевого базису.

**Обґрунтування варіанту**

Номер залікової книжки: 921910 = 100100000000112

X2 = –10010001,0000111

Y2 = +10100,0010000111

**Виконання роботи** (основна частина)

1. Числа  і  в прямому коді у формі з плаваючою комою:

P(X2ПК) = +810=10002 ;

M(X2ПК) = 1.100100010000111 ;

E (X2ПК) = 10111 ;

P(Y2ПК)=+510=1012 ;

M(Y2ПК)= 0.101000010000111 ;

E (Y2ПК) = 1100 .

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:

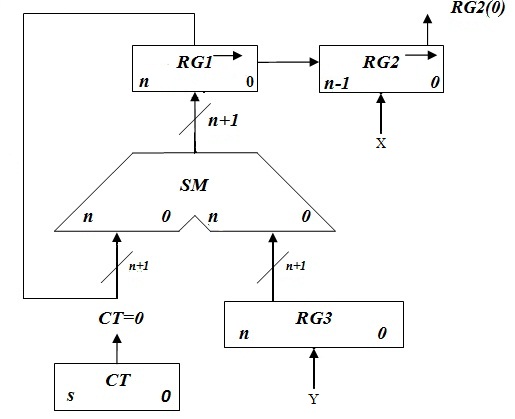
що рівносильно

.

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

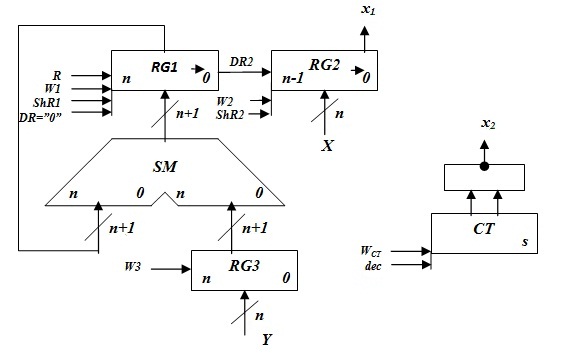
*Рисунок 2.1.2. Змістовний мікроалгоритм пристрою множення першим способом*

2.1.4 **Таблиця станів регістрів**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 | СТ |
| П.С. | 0000000000000000 | 10010001000011**1** | 0101000010000111 | 1111 |
| 1 | +0101000010000111  0101000010000111  0010100001000011 | 11001000100001**1** |  | 1110 |
| 2 | +0101000010000111  0111100011001010  0011110001100101 | 01100100010000**1** |  | 1101 |
| 3 | +0101000010000111  1000110011101100  0100011001110110 | 00110010001000**0** |  | 1100 |
| 4 | 0010001100111011 | 00011001000100**0** |  | 1011 |
| 5 | 0001000110011101 | 10001100100010**0** |  | 1010 |
| 6 | 0000100011001110 | 11000110010001**0** |  | 1001 |
| 7 | 0000010001100111 | 01100011001000**1** |  | 1000 |
| 8 | +0101000010000111  0101010011101110  0010101001110111 | 00110001100100**0** |  | 0111 |
| 9 | 0001010100111011 | 10011000110010**0** |  | 0110 |
| 10 | 0000101010011101 | 11001100011001**0** |  | 0101 |
| 11 | 0000010101001110 | 11100110001100**1** |  | 0100 |
| 12 | +0101000010000111  0101010111010101  0010101011101010 | 11110011000110**0** |  | 0011 |
| 13 | 0001010101110101 | 01111001100011**0** |  | 0010 |
| 14 | 0000101010111010 | 10111100110001**1** |  | 0001 |
| 15 | +0101000010000111  0101101101000001  **0, 010110110100000** | **110111100110001** |  | 0000 |

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 |  | RG2[0]  CT=0 | X1  X2 |
|
|
|  |

Початок

Кінець

R, W2, W3, WCT

W1

ShR1,ShR2,dec

X1

X2

1

1

Z1

Z2

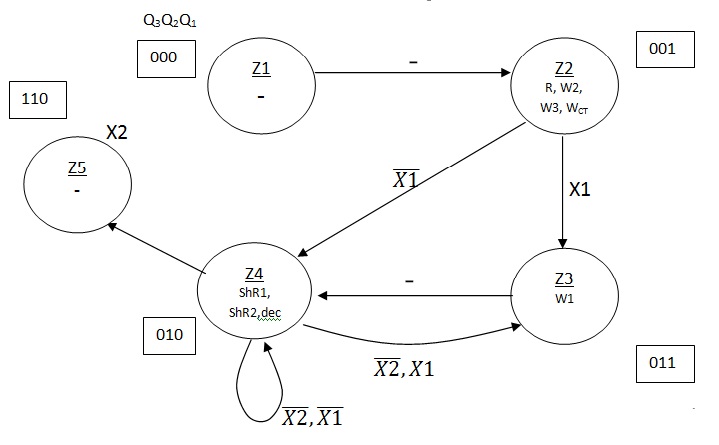
Z3

Z4

Z5

*Рисунок 2.1.4. Закодований мікроалгоритм пристрою множення першим способом*

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

**

*Рисунок 2.1.5. Граф автомата Мура пристрою множення першим способом*

2.5.7 **Обробка порядків**

Нормалізація мантиси:

MZ= ,010110110100000 <= ; ;

,101101101000000 ; .

Знак мантиси: .

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

Зн.РZ PZ=+1210 Зн.МZ MZ

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

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

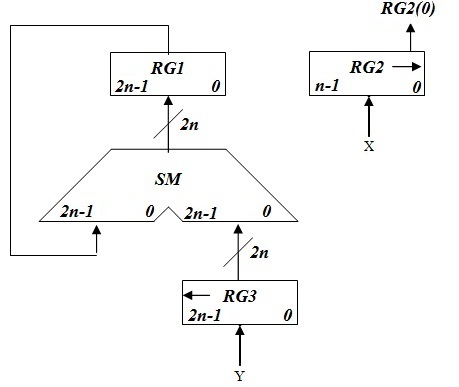
.

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

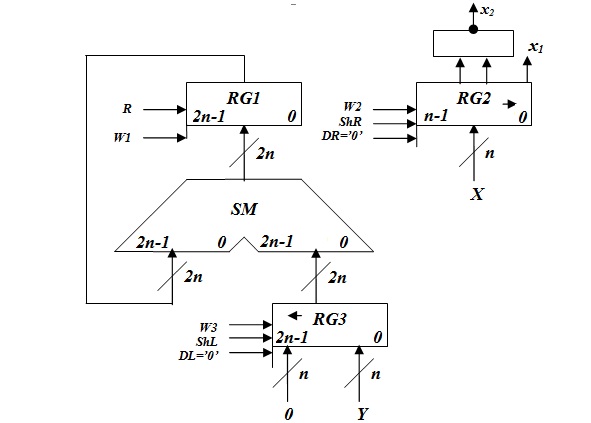
*Рисунок 2.2.2. Змістовний мікроалгоритм пристрою множення другим способом*

2.2.4. **Таблиця станів регістрів**

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

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG3 | RG2 |
| П.С. | 000000000000000000000000000000 | 000000000000000101000010000111 | 10010001000011**1** |
| 1 | 000000000000000000000000000000  +000000000000000101000010000111  000000000000000101000010000111 | 000000000000001010000100001110 | 01001000100001**1** |
| 2 | +000000000000001010000100001110  000000000000001111000110010101 | 000000000000010100001000011100 | 00100100010000**1** |
| 3 | +000000000000010100001000011100  000000000000100011001110110001 | 000000000000101000010000111000 | 00010010001000**0** |
| 4 | 000000000000100011001110110001 | 000000000001010000100001110000 | 00001001000100**0** |
| 5 | 000000000000100011001110110001 | 000000000010100001000011100000 | 00000100100010**0** |
| 6 | 000000000000100011001110110001 | 000000000101000010000111000000 | 00000010010001**0** |
| 7 | 000000000000100011001110110001 | 000000001010000100001110000000 | 00000001001000**1** |
| 8 | +000000001010000100001110000000  000000001010100111011100110001 | 000000010100001000011100000000 | 00000000100100**0** |
| 9 | 000000001010100111011100110001 | 000000101000010000111000000000 | 00000000010010**0** |
| 10 | 000000001010100111011100110001 | 000001010000100001110000000000 | 00000000001001**0** |
| 11 | 000000001010100111011100110001 | 000010100001000011100000000000 | 00000000000100**1** |
| 12 | +000010100001000011100000000000  000010101011101010111100110001 | 000101000010000111000000000000 | 00000000000010**0** |
| 13 | 000010101011101010111100110001 | 001010000100001110000000000000 | 00000000000001**0** |
| 14 | 000010101011101010111100110001 | 010100001000011100000000000000 | 00000000000000**1** |
| 15 | +010100001000011100000000000000  **010110110100000110111100110001** | 101000010000111000000000000000 | **000000000000000** |

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



*Рисунок 2.2.3. Функціональна схема пристрою множення другим способом*

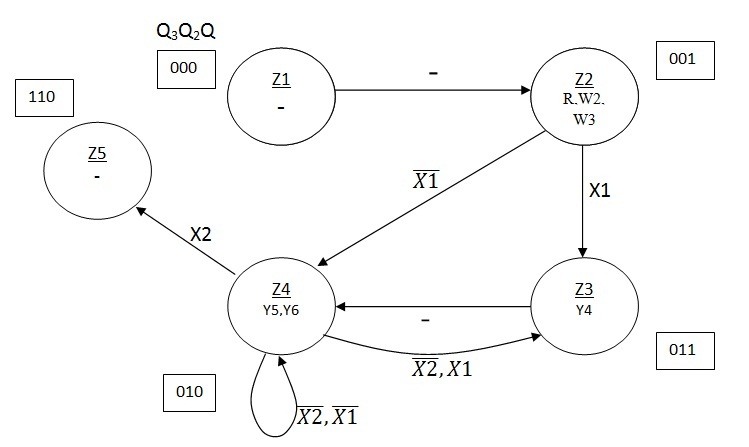
2.2.6 **Закодований мікроалгоритм**

*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**

Нормалізація мантиси:

MZ= ,010110110100000 <= ;

,101101101000000 ; .

Знак мантиси: .

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

Зн.РZ PZ=+1210 Зн.МZ MZ

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

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

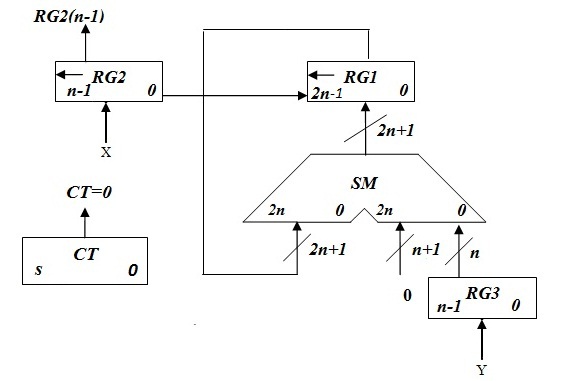
*.*

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

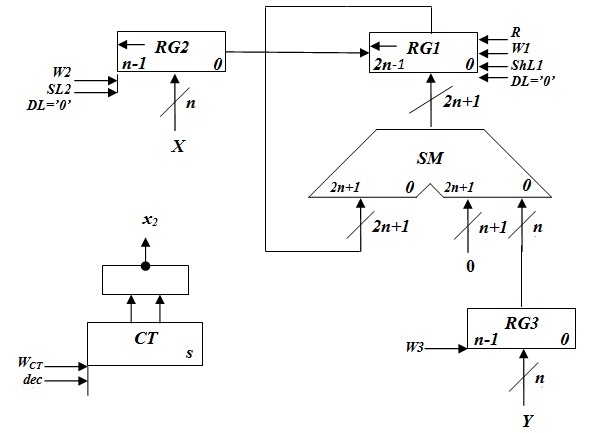
*Рисунок 2.3.2. Змістовний мікроалгоритм пристрою множення третім способом*

2.3.4 **Таблиця станів регістрів**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1 | RG3 | RG2 | СТ |
| П.С. | 000000000000000000000000000000 | 101000010000111 | **1**00100010000111 | 1111 |
| 1 | +000000000000000101000010000111  000000000000000101000010000111 |  | **0**01000100001110 | 1110 |
| 000000000000001010000100001110 |
| 2 | 000000000000010100001000011100 |  | **0**10001000011100 | 1101 |
| 3 | 000000000000101000010000111000 |  | **1**00010000111000 | 1100 |
| 4 | +000000000000000101000010000111  000000000000101101010010111111 |  | **0**00100001110000 | 1011 |
| 000000000001011010100101111110 |
| 5 | 000000000010110101001011111100 |  | **0**01000011100000 | 1010 |
| 6 | 000000000101101010010111111000 |  | **0**10000111000000 | 1001 |
| 7 | 000000001011010100101111110000 |  | **1**00001110000000 | 1000 |
| 8 | +000000000000000101000010000111  000000001011011001110001110111 |  | **0**00011100000000 | 0111 |
| 000000010110110011100011101110 |
| 9 | 000000101101100111000111011100 |  | **0**00111000000000 | 0110 |
| 10 | 000001011011001110001110111000 |  | **0**01110000000000 | 0101 |
| 11 | 000010110110011100011101110000 |  | **0**11100000000000 | 0100 |
| 12 | 000101101100111000111011100000 |  | **1**11000000000000 | 0011 |
| 13 | +000000000000000101000010000111  000101101100111101111101100111  001011011001111011111011001110 |  | **1**10000000000000 | 0010 |
| 14 | +000000000000000101000010000111  001011011010000000111101010101 |  | **1**00000000000000 | 0001 |
| 010110110100000001111010101010 |
| 15 | **+**000000000000000101000010000111  **,010110110100000110111100110001** |  | **0**00000000000000 | **0000** |

2.3.5 **Функціональна схема з відображенням управляючих сигналів**

**

*Рисунок 2.3.3 Функціональна схема пристрою множення третім способом*

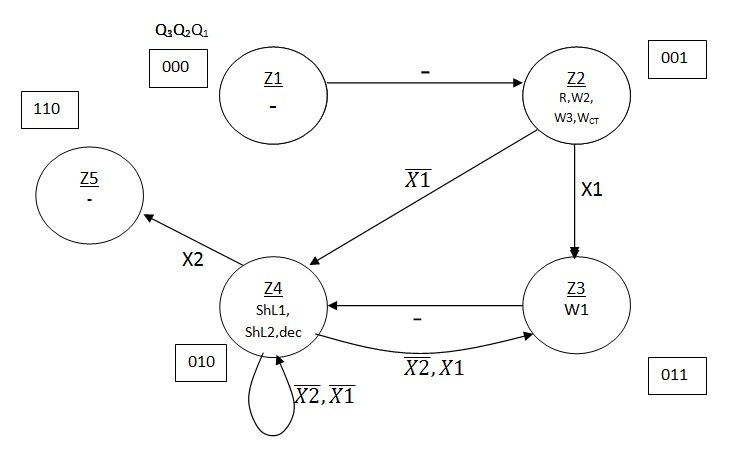
2.3.6 **Закодований мікроалгоритм**

*Таблиця 2.3.2 Таблиця кодування мікрооперацій пристрою множення третім способом*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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).0  CT:=CT-1 | R  W2  W3  WCT  W1  ShL1  ShL2  dec |  | RG2[n-1]  CT=0 | X1  X2 |
| Початок  Кінець  R,W2,W3,WCT  W1  ShL1,ShL2,dec  X1  X2  1  1  Z1  Z2  Z3  Z4  Z5 |

*Рисунок 2.3.4. Закодований мікроалгоритм пристрою множення третім способом*

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

**

*Рисунок 2.3.5. Граф автомата Мура пристрою множення третім способом*

2.3.8 **Обробка порядків**

Нормалізація мантиси:

MZ= ,010110110100000 <= ;

,101101101000000 ; .

Знак мантиси:

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

Зн.РZ PZ=+1210 Зн.МZ MZ

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

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

2.4 **Четвертий спосіб множення**

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

Вираз

*подамо у вигляді*

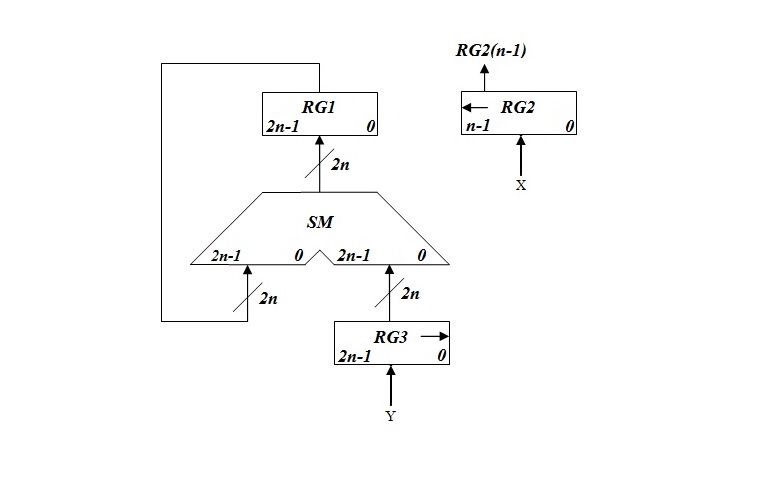
*.*

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

.

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

2.4.2 **Операційна схема**

****

*Рисунок 2.4.1. Операційна схема пристрою множення четвертим способом*

2.4.3 **Змістовний мікроалгоритм**

Початок

Кінець

RG1:=0 ; RG2:=X

RG3:=Y

RG3:=0.r(RG3)

RG1:=RG1+RG3

RG3:=0.r(RG3)

RG2:=l(RG2).0

RG2[n-1]

RG2=0

1

1

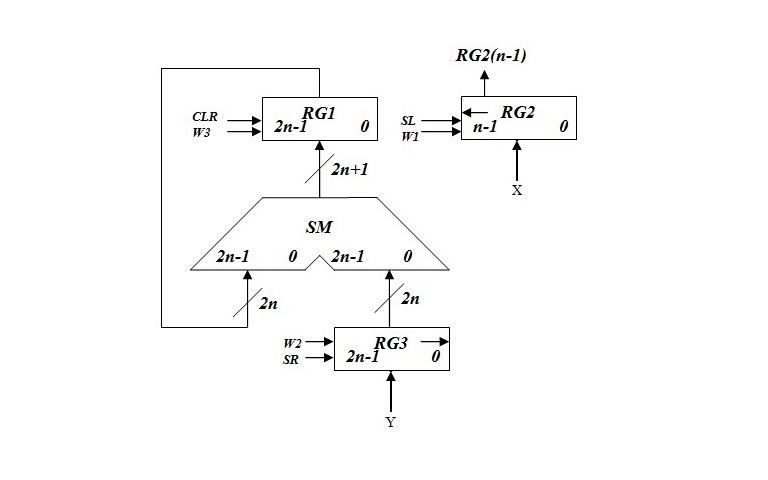
*Рисунок 2.4.2. Змістовний мікроалгоритм пристрою множення четвертим способом*

2.4.4 **Таблиця станів регістрів**

*Таблиця 2.4.1. Таблиця станів регістрів пристрою множення четвертим способом*

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG3 | RG2 |
| П.С. | 000000000000000000000000000000 | 101000010000111000000000000000  010100001000011100000000000000 | **1**00100010000111 |
| 1 | +010100001000011100000000000000  010100001000011100000000000000 | 001010000100001110000000000000 | **0**01000100001110 |
| 2 | 010100001000011100000000000000 | 000101000010000111000000000000 | **0**10001000011100 |
| 3 | 010100001000011100000000000000 | 000010100001000011100000000000 | **1**00010000111000 |
| 4 | +000010100001000011100000000000  010110101001011111100000000000 | 000001010000100001110000000000 | **0**00100001110000 |
| 5 | 010110101001011111100000000000 | 000000101000010000111000000000 | **0**01000011100000 |
| 6 | 010110101001011111100000000000 | 000000010100001000011100000000 | **0**10000111000000 |
| 7 | 010110101001011111100000000000 | 000000001010000100001110000000 | **1**00001110000000 |
| 8 | +000000001010000100001110000000  010110110011100011101110000000 | 000000000101000010000111000000 | **0**00011100000000 |
| 9 | 010110110011100011101110000000 | 000000000010100001000011100000 | **0**00111000000000 |
| 10 | 010110110011100011101110000000 | 000000000001010000100001110000 | **0**01110000000000 |
| 11 | 010110110011100011101110000000 | 000000000000101000010000111000 | **0**11100000000000 |
| 12 | 010110110011100011101110000000 | 000000000000010100001000011100 | **1**11000000000000 |
| 13 | +000000000000010100001000011100  010110110011110111110110011100 | 000000000000001010000100001110 | **1**10000000000000 |
| 14 | +000000000000001010000100001110  010110110100000001111010101010 | 000000000000000101000010000111 | **1**00000000000000 |
| 15 | +000000000000000101000010000111  **,010110110100000110111100110001** | 000000000000000010100001000011 | **000000000000000** |

*2.4.5* **Функціональна схема з відображенням управляючих сигналів**

**

*Рисунок 2.4.3. Функціональна схема пристрою множення четвертим способом*

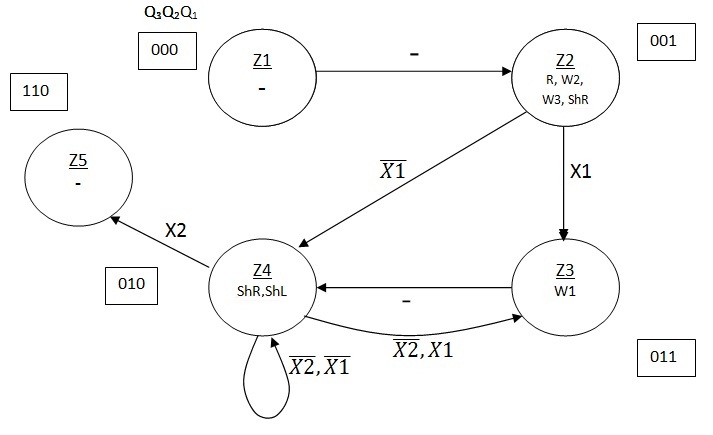
2.4.6 **Закодований мікроалгоритм**

*Таблиця 2.4.2. Таблиця кодування мікрооперацій пристрою множення четвертим способом*

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

*Рисунок 2.4.4. Закодований мікроалгоритм пристрою множення четвертим способом*

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

**

*Рисунок 2.4.5. Граф автомата Мура пристрою множення четвертим способом*

2.4.8 **Обробка порядків**

Нормалізація мантиси:

MZ= ,010110110100000 <= ;

101101101000000 ; .

Знак мантиси:

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

Зн.РZ PZ=+1210 Зн.МZ MZ

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

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

2.5 **Перший спосіб ділення (з відновленням від’ємного залишку)**

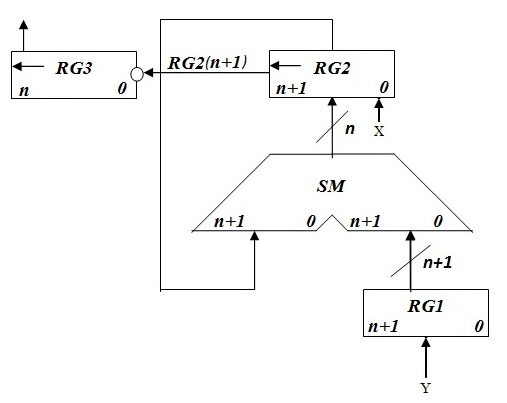
2.5.1 **Теоретичне обґрунтування способу**

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.

2.5.2 **Операційна схема**

***RG3(n)***



*Рисунок 2.5.1. Операційна схема пристрою ділення першим способом*

2.5.3 **Змістовний мікроалгоритм**

Початок

Кінець

RG3:=l(RG3).

RG2:=l(RG2).0

RG2:=RG2++1

RG2:=RG2+RG1

RG2[n+1]

RG3[n+1]

1

1

RG3:=0

RG2:=X;

RG1:=Y;

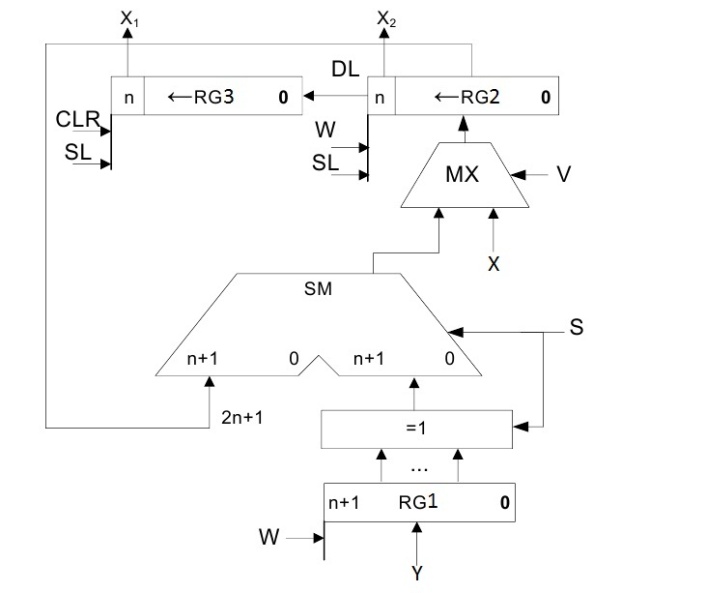
*Рисунок 2.5.2. Змістовний мікроалгоритм пристрою ділення першим способом*

2.5.4 **Таблиця станів регістрів**

*Таблиця 2.5.1. Таблиця станів регістрів пристрою ділення першим способом*

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG3 | RG2 | RG1 |
| П.С. | 0000000000000000 | 00100100010000111 | 00101000010000111пк  11010111101111001дк |
| 1 | 000000000000000**1** | **0**1001000100001110  +11010111101111001  00100000010000111 |  |
| 2 | 00000000000000**1**1 | **0**1000000100001110  +11010111101111001  00011000010000111 |  |
| 3 | 0000000000000**1**11 | **0**0011000010000111  +11010111101111001  00001000010000111 |  |
| 4 | 000000000000**1**111 | **0**0010000100001110  +11010111101111001  11101000010000111 |  |
| 5 | 00000000000**1**1110 | **1**1010000100001110  +00101000010000111  11111000110010101 |  |
| 6 | 0000000000**1**11100 | **1**1110001100101010  +00101000010000111  00011001110110001 |  |
| 7 | 000000000**1**111001 | **0**0110011101100010  +11010111101111001  00001011011011011 |  |
| 8 | 00000000**1**1110011 | **0**0010110110110110  +11010111101111001  11101110100101111 |  |
| 9 | 0000000**1**11100110 | **1**1011101001011110  +00101000010000111  00000101011100101 |  |
| 10 | 000000**1**111001101 | **0**0001010111001010  +11010111101111001  11100010101000011 |  |
| 11 | 00000**1**1110011010 | **1**1000101010000110  +00101000010000111  11101101100001101 |  |
| 12 | 0000**1**11100110100 | **1**1011011000011010  +00101000010000111  00000011010100001 |  |
| 13 | 000**1**111001101001 | **0**0000110101000010  +11010111101111001  11011110010111011 |  |
| 14 | 00**1**1110011010010 | **1**0111100101110110  +00101000010000111  11100100111111101 |  |
| 15 | 0**1**11100110100100 | **1**1001001111111010  +00101000010000111  11110010010000001 |  |
| 16 | **1**,**111001101001000** | **1**1100100100000010  +00101000010000111  00001100110001001 |  |

2.5.5 **Функціональна схема з відображенням управляючих сигналів**

****

*Рисунок 2.5.3. Функціональна схема пристрою ділення першим способом*

2.5.6 **Закодований мікроалгоритм**

*Таблиця 2.5.2. Таблиця кодування мікрооперацій пристрою ділення першим способом*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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+1] | | X1  X2 |
|  | |

1

1

Початок

Кінець

ShL1, ShL2

W4

W5, S

X1

X2

W3, W2, W1, V

Z1

Z2

Z3

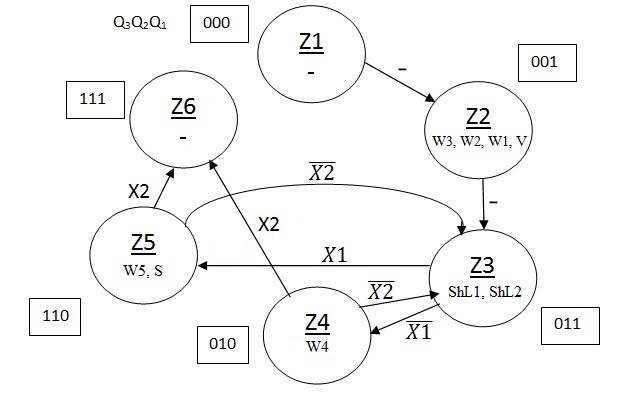
Z4

Z6

Z5

*Рисунок 2.5.4. Закодований мікроалгоритм пристрою ділення першим способом*

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

**

*Рисунок 2.5.5. Граф автомата Мура пристрою ділення першим способом*

2.5.8 **Обробка порядків і нормалізація**

.

Нормалізація мантиси не потрібна.

MZ= ,111001101001000.

Знак мантиси: .

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

Зн.РZ PZ=+1210 Зн.МZ MZ

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

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

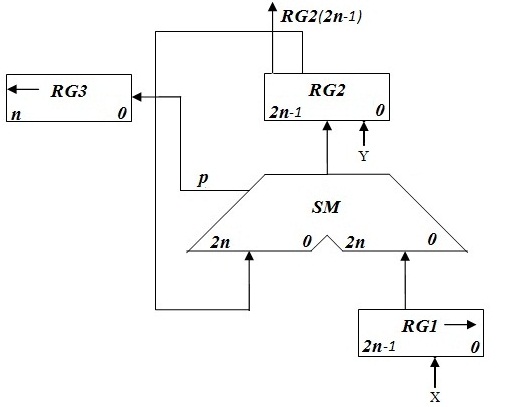
2.6 **Другий спосіб ділення (без відновлення остачі)**

2.6.1 **Теоретичне обґрунтування способу**

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 2.6.1. Операційна схема пристрою ділення другим способом*

2.6.3 **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-1]

RG3[n]

1

1

RG3:=0…01;

RG1:=Y;

RG2:= X

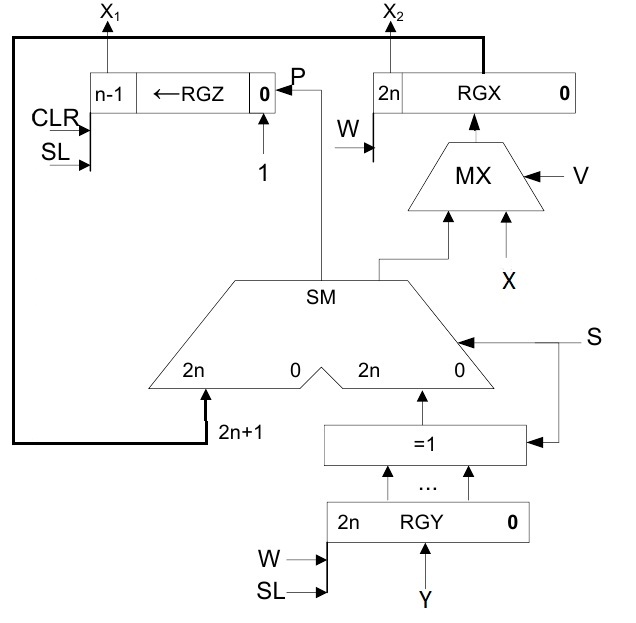
*Рисунок 2.6.2. Змістовний мікроалгоритм пристрою ділення другим способом*

2.6.4 **Таблиця станів регістрів**

*Таблиця 2.6.1. Таблиця станів регістрів пристрою ділення другим способом*

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG3 | RG2 | RG1 |
| П.С. | 000000000000000**1** | 010010001000011100000000000000 | 00.1010000100001110000000000000 |
| 1 | 00000000000000**1**1 | +110101111011110010000000000000  **0**01000000100001110000000000000 | 000101000010000111000000000000 |
| 2 | 0000000000000**1**11 | +111010111101111001000000000000  **0**00011000010000111000000000000 | 000010100001000011100000000000 |
| 3 | 000000000000**1**111 | +111101011110111100100000000000  000000100001000011100000000000 | 000001010000100001110000000000 |
| 4 | 00000000000**1**1110 | +111110101111011110010000000000  **1**11111010000100001110000000000 | 000000101000010000111000000000 |
| 5 | 0000000000**1**11100 | +000000101000010000111000000000  **1**11111111000110010101000000000 | 000000010100001000011100000000 |
| 6 | 000000000**1**111001 | +000000010100001000011100000000  **0**00000001100111011000100000000 | 000000001010000100001110000000 |
| 7 | 00000000**1**1110011 | +111111110101111011110010000000  **0**00000000010110110110110000000 | 000000000101000010000111000000 |
| 8 | 0000000**1**11100110 | +111111111010111101111001000000  **1**11111111101110100101111000000 | 000000000010100001000011100000 |
| 9 | 000000**1**111001101 | +000000000010100001000011100000  **0**00000000000010101110010100000 | 000000000001010000100001110000 |
| 10 | 00000**1**1110011010 | +111111111110101111011110010000  **1**11111111111000101010000110000 | 000000000000101000010000111000 |
| 11 | 0000**1**11100110100 | +000000000000101000010000111000  **1**11111111111101101100001101000 | 000000000000010100001000011100 |
| 12 | 000**1**111001101001 | +000000000000010100001000011100  **0**00000000000000001101010000100 | 000000000000001010000100001110 |
| 13 | 00**1**1110011010010 | +111111111111110101111011110010  **1**11111111111110111100101110110 | 000000000000000101000010000111 |
| 14 | 0**1**11100110100100 | +000000000000000101000010000111  **1**11111111111111100100111111101 | 000000000000000010100001000011 |
| 15 | **1,111001101001000** | +000000000000000010100001000011  **1**11111111111111111001001000000 | 000000000000000001010000100001 |

2.6.5 **Функціональна схема з відображенням управляючих сигналів**



*Рисунок 2.6.3. Функціональна схема пристрою ділення другим способом*

*2.6.6* **Закодований мікроалгоритм**

*Таблиця 2.6.2. Таблиця кодування мікрооперацій пристрою ділення другим способом*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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  W3  ShR  ShL  W4 |  | RG2[2n+1]  RG3[n] | X1  X2 |
|  |

1

1

Початок

Кінець

W3,ShR,ShL, S

W4,ShR,ShL

X1

X2

R,W1,W2,V

Z1

Z2

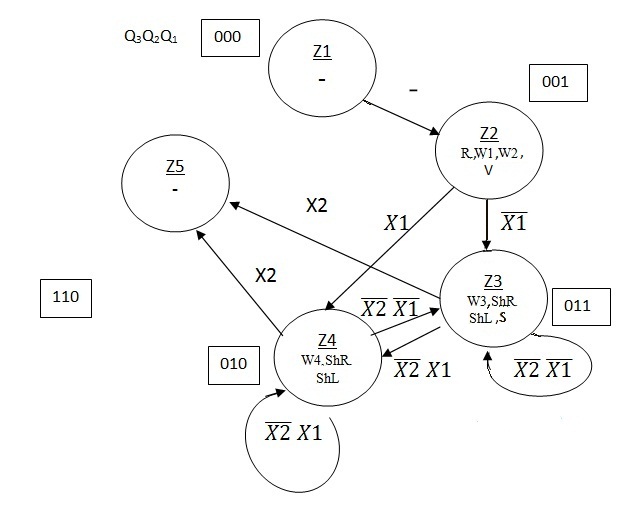
Z3

Z4

Z5

*Рисунок 2.6.4. Закодований мікроалгоритм пристрою ділення другим способом*

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



*Рисунок 2.6.5 Граф автомата Мура пристрою ділення другим способом*

*2.6.8* **Обробка порядків і нормалізація**

.

Нормалізація мантиси не потрібна.

MZ= ,111001101001000.

Знак мантиси: .

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

Зн.РZ PZ=+1210 Зн.МZ MZ

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

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

2.7 **Операція додавання чисел**

Z=X+Y.

2.7.1 **Теоретичне обґрунтування**

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.

**Вирівнювання порядків:**

,

.

Робимо зсув вправо MY, зменшуючи на кожному кроці, доки стане 0.

*Таблиця 2.7.1. Таблиця зсуву мантиси на етапі вирівнювання порядків при додаванні чисел*

|  |  |  |
| --- | --- | --- |
|  |  | Мікрооперація |
| 0.101000010000111 | 11 | П.С. |
| 0.010100001000011 | 10 | 🡪 |
| 0.001010000100001 | 01 | 🡪 |
| 0.000101000010000 | 00 | 🡪 |

**Додавання мантис у модифікованому ДК**

*Таблиця 2.7.2. Додавання мантис*

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

*.*

2.7.2 **Операційна схема**

m- розрядність мантиси

n- розрядність порядку

q=]log2m[

R

L

n+1 RGPZ 0

m+1 RGZ 0

*КС*

q CT 1

CT=0

n

*Рисунок 2.7.1. Операційна схема пристрою нормалізації при додавання чисел*

Виконаємо синтез комбінаційної схеми для визначення порушення нормалізації.

*Таблиця 2.7.3. Визначення порушення нормалізації при додаванні чисел*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Розряди регістру RGZ | | | Значення функцій | |
| Z’0 | Z0 | Z1 | L | R |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |

L= Z0, R=.

Результат беремо по модулю, знак встановлюємо за Z’0 до нормалізації.

2.7.3 **Змістовний мікроалгоритм**

Початок

Кінець

0

RGZ:=RGZ(m+2).r(RGZ)

RGPZ:=RGPZ+1

RGZ:=l(RGZ).0

RGPZ:=RGPZ-1

CT:=CT-1;

CT:=m; RGZ:=Z;

1

1

1

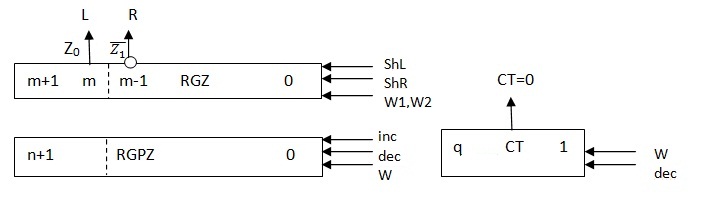
*Рисунок 2.7.2. Змістовний мікроалгоритм пристрою додавання двох чисел*

*2.7.4* **Таблиця станів регістрів**

*Таблиця 2.7.4. Таблиця станів регістрів пристрою додавання двох чисел*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ циклу** | **RGPZ** | **RGZ** | **L** | **R** | **СT** | **МО** |
| **ПС** | 001000 | **1**1,011111010101110 | 0 | 1 | 100 |  |
| **1** | 000111 | 0**0**,**0**11111010101110  0**0**,**1**11110101011100 | 0 | 0 | 011 | Z’0 Z0:=  RGZ:=l(RGZ).0  RGPZ:=RGPZ-1  CT:=CT-1 |

2.7.5 **Функціональна схема з відображенням управляючих сигналів**

**

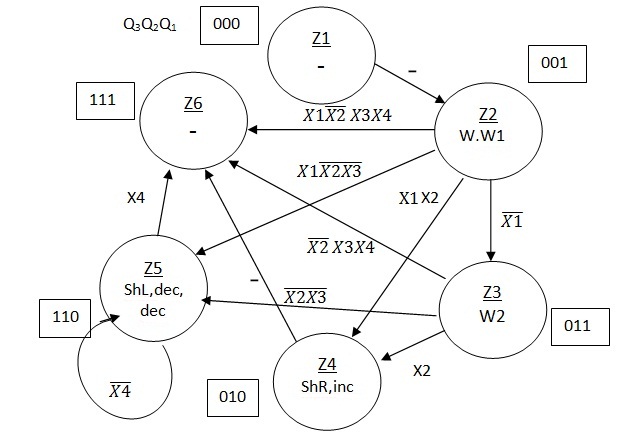
*Рисунок 2.7.4. Функціональна схема пристрою додавання двох чисел*

*2.7.6* **Закодований мікроалгоритм**

*Таблиця 2.7.5. Таблиця кодування мікрооперацій пристрою додавання двох чисел*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |
| *Рисунок 2.7.5 Мікроалгоритм пристою*  *додавання*  Початок  Кінець  ShR,inc  ShL,dec,dec  W,W1  X4  1  1  1  X1  W2  1  Z1  Z2  Z3  Z4  Z5  Z6 |

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

****

*Рисунок 2.7.6. Граф автомата Мура пристрою додавання двох чисел*

2.7.8 **Обробка порядків**

PZ=111

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

Зн.РZ PZ=+710 Зн.МZ  MZ

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

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

2.8 **Операція добування кореня**

Z=

2.8.1 **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.

;

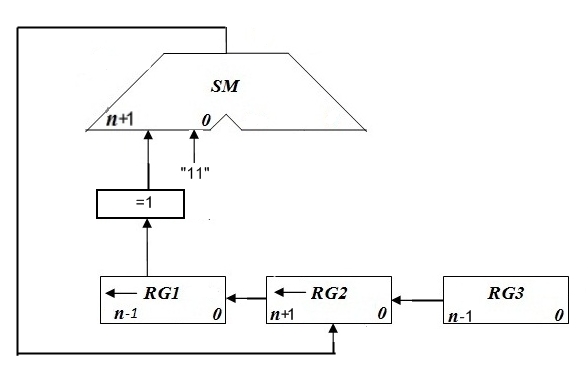
2. Якщо , то .

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.

2.8.2 **Операційна схема**

****

***n***

***0***

***S***

***СТ=0***

СТ

***n***

***n***

*Рисунок 2.8.1. Операційна схема пристрою добування кореня числа*

2.8.3 **Змістовний мікроалгоритм**

Початок

Кінець

R2:=R2+R1.11

R2:=R2+.11

R2[n+1]

СТ=0

1

1

R3:=X;

R2:=0;

R1:=0

CT:=n

R1:=l[R1].

CT:=CT-1

R2=l2[R2].R3(n-1).R3(n-2)

R3:=l2[R3].00

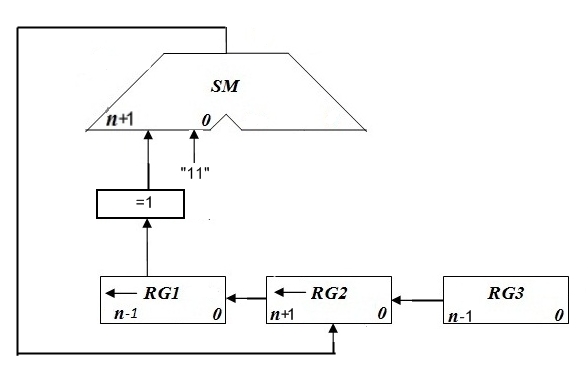
*Рисунок 2.8.2. Змістовний мікроалгоритм пристрою добування кореня числа*

2.8.4 **Таблиця станів регістрів**

*Таблиця 2.8.1 Таблиця станів регістрів пристрою добування кореня числа*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | R1 | R2 | R3 | CT |
| П.С. | 000000000000000 | 00000000000000000 | 100100010000111 | 1111 |
| 1 | 000000000000001 | **0**0000000000000010  +11111111111111111  00000000000000001 | 010001000011100 | 1110 |
| 2 | 000000000000011 | **0**0000000000000101  +11111111111111011  00000000000000000 | 000100001110000 | 1101 |
| 3 | 000000000000110 | **0**0000000000000000  +11111111111110011  11111111111110011 | 010000111000000 | 1100 |
| 4 | 000000000001100 | **1**1111111111001101  +00000000000011011  11111111111101000 | 000011100000000 | 1011 |
| 5 | 000000000011000 | **1**1111111110100000  +00000000000110011  11111111111010011 | 001110000000000 | 1010 |
| 6 | 000000000110000 | **1**1111111101001100  +00000000001100011  11111111110101111 | 111000000000000 | 1001 |
| 7 | 000000001100000 | **1**1111111010111111  +00000000011000011  11111111110000010 | 100000000000000 | 1000 |
| 8 | 000000011000000 | **1**1111111000001010  +00000000110000011  11111111110001101 | 000000000000000 | 0111 |
| 9 | 000000110000001 | **1**1111111000110100  +00000001100000011  00000000100110111 | 000000000000000 | 0110 |
| 10 | 000001100000010 | **0**0000010011011100  +11111100111111011  11111111011010111 | 000000000000000 | 0101 |
| 11 | 000011000000101 | **1**1111101101011100  +00000110000001011  00000011101100111 | 000000000000000 | 0100 |
| 12 | 000110000001011 | **0**0001110110011100  +11110011111101011  00000010110000111 | 000000000000000 | 0011 |
| 13 | 001100000010110 | **0**0001011000011100  +11100111111010011  11110010111101111 | 000000000000000 | 0010 |
| 14 | 011000000101100 | **1**1001011110111100  +00110000001011011  11111100000010111 | 000000000000000 | 0001 |
| 15 | **110000001011001** | **1**1110000001011100  +01100000010110011  01010000100001111 | 000000000000000 | 0000 |

2.8.5 **Функціональна схема з відображенням управляючих сигналів**



***S***

***n***

***n***

***q***

***0***

*Рисунок 2.8.3. Функціональна схема пристрою добування кореня числа*

R

W1

W2

ShLL

R1

ShL

R

ShLL0

***СТ=0***

СТ

***n***

*2.8.6* **Закодований мікроалгоритм**

*Таблиця 2.8.2. Таблиця кодування мікрооперацій пристрою добування кореня числа*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| RX:=X;  RR:=0;  RZ:=0  CT:=15  RR:=RR+RZ.11  RR:=RR+.11  RR=LL(RR).RX(n-1;n-2)  RX:=LL(RX).00  RZ:=L(RZ).  CT:=CT-1 | WX  R  R1  WCT  W1  W2  ShLL  ShLL0  ShL  dec |  | RR[n+1]  RZ[n] | X1  X2 |
|  |

Кінець

W1

W2,S

X1

X2

1

1

ShLL,ShLL0,ShL,dec

Початок

WX,R,R1,WCT

Z1

Z2

Z4

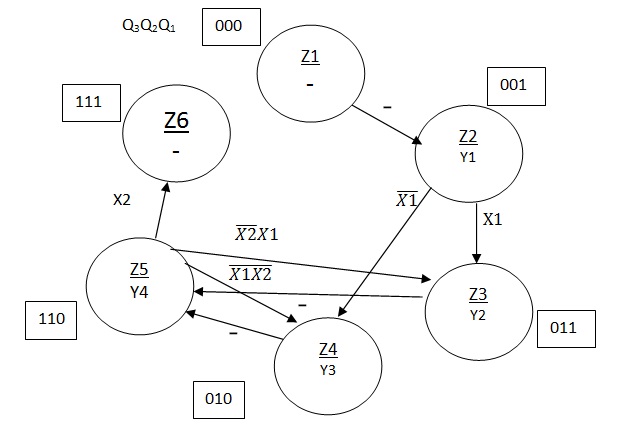
Z3

Z5

Z6

*Рисунок 2.8.4. Закодований мікроалгоритм пристрою добування кореня числа*

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

**

**,S**

*Рисунок 2.8.5. Граф автомата Мура пристрою добування кореня числа*

2.8.8 **Обробка порядків**

P1=P2:2=8:2=410=1002.

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

Зн.РZ PZ=+410 Зн.МZ MZ

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

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

3.**Синтез управляючого автомату Мура**

x3x2x1=0112=310 – операція множення третім способом.

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

За закодованим мікроалгоритмом (Рис. 2.3.4.) складемо таблицю:

*Таблиця 3.1 Таблиця кодування сигналів пристрою множення третім способом*

|  |  |
| --- | --- |
| Входи операційного автомата | Виходи управляючого автомата |
| R,W2,W3,WCT | Y1 |
| W1 | Y2 |
| ShL1,ShL2,dec | Y3 |

3.2 **Мікроалгоритм в термінах управляючого автомата**

Зробимо автомат Мура циклічним для спрощення будови автомату.

Початок

Кінець

Y1

Y2

Y3

X1

X2

1

1

Z1

Z2

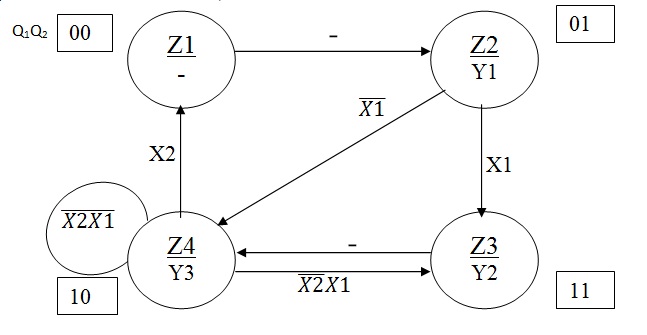
Z3

Z4

Z1

*Рисунок 3.1. Закодований мікроалгоритм пристрою множення третім способом*

Будуємо граф автомата Мура



*Рисунок 3.2. Граф автомата Мура пристрою множення третім способом*

3.3 **Структурна таблиця автомата**

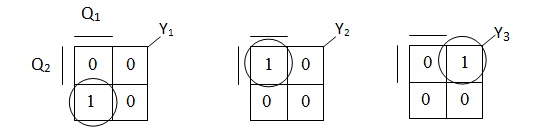
За графом автомата мура складаємо структурну таблицю автомата. Значення функцій збудження тригерів визначимо відповідно до графічної схеми переходів JK-тригера.

*Таблиця 3.2. Структурна таблиця автомата пристрою множення третім способом*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Перехід | Q2Q1 | Q2Q1 | x1x2 | y1y2y3 | J2 | K2 | J1 | K1 |
| *z1z2* | 0 0 | 0 1 | - - | 0 0 0 | 0 | - | 1 | - |
| *z2 z3* | 0 1 | 1 1 | 1 - | 1 0 0 | 1 | - | - | 0 |
| *z2 z4* | 0 1 | 1 0 | 0 - | 1 0 0 | 1 | - | - | 1 |
| *z3 z4* | 1 1 | 1 0 | - - | 0 1 0 | - | 0 | - | 1 |
| *z4 z3* | 1 0 | 1 1 | 1 0 | 0 0 1 | - | 0 | 1 | - |
| *z4 z1* | 1 0 | 0 0 | - 1 | 0 0 1 | - | 1 | 0 | - |
| *z4 z4* | 1 0 | 1 0 | 0 0 | 0 0 1 | - | 0 | 0 | - |

JK-тригер:

3.4 **Синтез функцій виходів і переходів**



Q1

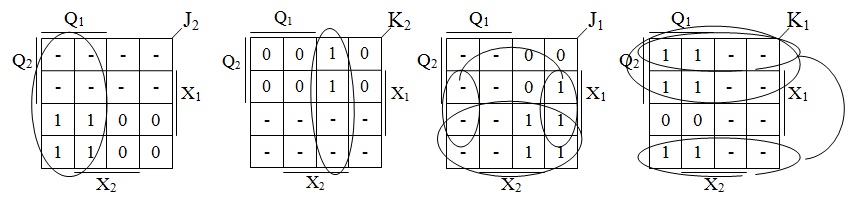
Q2

Q2

Q1

Q1

Q2



*Рисунок 3.3. Діаграми Вейча функцій виходів і переходів*

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

**підключені до входів операційного автомата)**

R

УПС

J

J1

&

1

&

1

K2

K1

K2

K

T2

C

G

S

R

J

J1

T1

K1

K

C

S

***2n+1***

***2n+1***

***0***

***0***

***SM***

***n***

***2n+1***

***n+1***

**0**

***RG2***

***0***

***n-1***

***RG1***

***2n***

***0***

***2n+1***

***RG3***

***0***

***n-1***

***Y1***

***Y2***

***Y3***

***Y1***

***Y3***

***Y1***

***n***

***X***

***n***

***Y***

***CT***

***s***

***Y1***

***Y3***

***x2***

*Рисунок 3.4. Функціональна схема пристрою множення третім способом*

**Висновок**

Під час виконання розрахункової роботи я виконав вісім арифметичних операцій над заданими згідно з варіантом завдання числами. При цьому я подав теоретичне обґрунтування кожної операції, операційні схеми пристроїв, що реалізують операції, їх змістовні алгоритми, таблиці станів регістрів, закодовані мікроалгоритми та управляючі автомати Мура. Також я представив обробку порядків результату кожної операції, нормалізацію їх мантис та форму запису результатів в пам'ять. Крім цього я синтезував управляючий автомат Мура на JK-тригерах для операції множення третім способом та побудував функціональну схему цього автомату.