НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

# "КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

#### ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

### Кафедра обчислювальної техніки

## РОЗРАХУНКОВА РАБОТА

з дисципліни "Комп’ютерна арифметика"

Виконав: Кузьмич Валентин Анатолійович

Група ІО-31,

Факультет ІОТ,

Залікова книжка № 3115

Допущена до захисту \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Номер технічного завдання 110000101011

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис керівника)

**I. 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 додатнього числа). 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 управл. 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 +1 побудувати управляючий автомат Мура на тригерах (тип вибрати самостійно) і елементах булевого базису.

**II.Обгрунтування варіанту:**

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

 і ,

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

 =-10010011,0100111

 =+10100,0110100111

**III.Ocновна частина:**

**Завдання 1**

=1.10010011,0100111;

=0.10100,0110100111;

Представлення чисел у формі з плаваючою точкою з порядком і мантисою:

Px=+10002; Mx=-1001001101001112;

Py=+01012; My=+101000110100111;

X2:

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

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

Y2:

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

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

Представлення чисел у формі з плаваючою точкою з харктеристикою і мантисою:

Ex = Px+2m = 810+27 = 810+12810 = 13610  = 100010002; Mx==-1001001101001112;

Ey = Py+2m = 510+27 = 510 + 12810 = 13310 = 100001012; My=+1010001101001112;

X2:

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

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

Y2:

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

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

**Завдання 2**

**2.1 Перший спосіб множення.**

**2.1.1 Теоретичне обґрунтування першого способу множення:**

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.

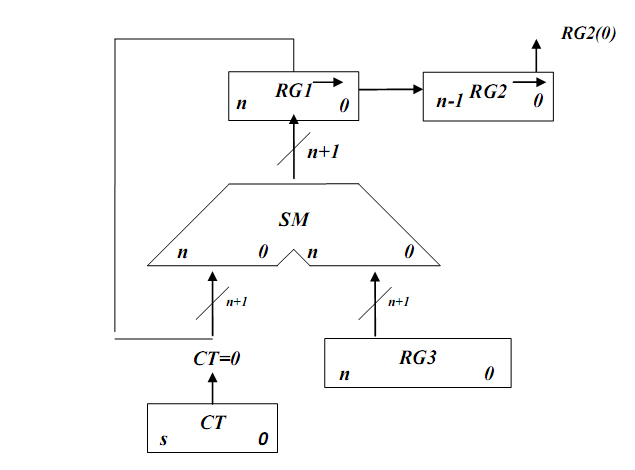
Множення мантис першим способом здійснюється з молодших розрядів множника, сума часткових добутків зсувається вправо, а множене залишається нерухомим. Тоді добуток двох чисел представляється у вигляді:

Z=Y+ Y…+ Y;

Z=(((0+Y)+ Y)…+ Y);

Z=;

**2.1.2 Операційна схема:**



*Рисунок 2.1.1. Операційна схема пристрою множення першим способом*

**2.1.3 Змістовний мікроалгоритм:**

Початок

RG1:=0; RG2:=X RG3:=Y; CT:=15

0

RG2[0]

1

RG1:=RG1+RG3

RG1:=0.r(RG1) RG2:=RG1[0].r(RG2) CT:=CT-1

0

CT=0

Кінець

1

*Рисунок 2.1.2. Змістовний мікроалгоритм пристрою множення першим способом*

**2.1.4 Таблиця станів регістрів:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | RG1 | RG2 | RG3 | CT |
| ПС | 0 | 100100110100111 | 101000110100111 | 1111 |
| 1 | 0010100011010011 | 110010011010011 | 101000110100111 | 1110 |
| 2 | +  0101000110100111  =  0111101001111010  0011110100111101 | 011001001101001 | 101000110100111 | 1101 |
| 3 | +  0101000110100111  =  1000111011100100  0100011101110010 | 001100100110100 | 101000110100111 | 1100 |
| 4 | 0010001110111001 | 000110010011010 | 101000110100111 | 1011 |
| 5 | 0001000111011100 | 100011001001101 | 101000110100111 | 1010 |
| 6 | +  0101000110100111  =  0110001110000011  0011000111000001 | 110001100100110 | 101000110100111 | 1001 |
| 7 | 0001100011100000 | 111000110010011 | 101000110100111 | 1000 |
| 8 | +  0101000110100111  =  0110101010000111  0011010101000011 | 111100011001001 | 101000110100111 | 0111 |
| 9 | +  0101000110100111  =  1000011011101010  0100001101110101 | 011110001100100 | 101000110100111 | 0110 |
| 10 | 0010000110111010 | 101111000110010 | 101000110100111 | 0101 |
| 11 | 0001000011011101 | 010111100011001 | 101000110100111 | 0100 |
| 12 | +  0101000110100111  =  0110001010000100  0011000101000010 | 001011110001100 | 101000110100111 | 0011 |
| 13 | 0001100010100001 | 000101111000110 | 101000110100111 | 0010 |
| 14 | 0000110001010000 | 100010111100011 | 101000110100111 | 0001 |
| 15 | +  0101000110100111  =  0101110111110111  0010111011111011 | 110001011110001 | 101000110100111 | 0000 |

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

***x1***

***DR2***

***R***

***W1***

***ShR1***

***DR=”0”***

***RG2***

***0***

***n-1***

***RG1***

***0***

***n***

***W2***

***ShR2***

***n***

***n+1***

***x2***

***X***

***n***

***n***

***0***

***0***

***SM***

***n+1***

***n+1***

***CT***

***s***

***WCT***

***dec***

***W3***

***RG3***

***0***

***n***

***n***

***Y***

*Рисунок 2.1.3. Функціональна схема пристрою множення першим способом*

**2.1.6 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 |

Z1

Z2

R, W2, W3, WCT

1

X1

Z3

W1

Z4

X2

ShR1,ShR2,dec

1

Кінець

Z5

*Рисунок 2.1.4. Закодований мікроалгоритм пристрою множення першим способом*

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

Z2

R, W2, W3, WCT

001

000

Q3Q2Q1

-

Z1

-

110

X2

Z5

**-**

X1

Z4

ShR1, ShR2,dec

-

Z3

W1

010

011

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

**2.1.8 Обробка порядків і нормалізація**

.

Отримали результат: Mz = 0,010111011111011 <=; ;

Mz = 0,101110111110110, ; .

Знак мантиси: .

**2.1.9 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 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |

**2.2 Другий спосіб множення**

**2.2.1 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.*

**2.2.2 Операційна схема**

***RG2(0)***

***n-1***

***0***

***RG2***

***2n***

***RG1***

***0***

***2n-1***

***2n-1***

***2n-1***

***0***

***0***

***SM***

***2n***

***RG3***

***0***

***2n-1***

*Рисунок 2.2.1. Операційна схема пристрою множення другим способом*

**2.2.3 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 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 | 000000000000000101000110100111 | 100100110100111 |
| 1 | 000000000000000101000110100111 | 000000000000001010001101001110 | 010010011010011 |
| 2 | +  000000000000001010001101001110  =  000000000000001111010011110101 | 000000000000010100011010011100 | 001001001101001 |
| 3 | +  000000000000010100011010011100  =  000000000000100011101110010001 | 000000000000101000110100111000 | 000100100110100 |
| 4 | 000000000000100011101110010001 | 000000000001010001101001110000 | 000010010011010 |
| 5 | 000000000000100011101110010001 | 000000000010100011010011100000 | 000001001001101 |
| 6 | +  000000000010100011010011100000  =  000000000011000111000001110001 | 000000000101000110100111000000 | 000000100100110 |
| 7 | 000000000011000111000001110001 | 000000001010001101001110000000 | 000000010010011 |
| 8 | +  000000001010001101001110000000  =  000000001101010100001111110001 | 000000010100011010011100000000 | 000000001001001 |
| 9 | +  000000010100011010011100000000  =  000000100001101110101011110001 | 000000101000110100111000000000 | 000000000100100 |
| 10 | 000000100001101110101011110001 | 000001010001101001110000000000 | 000000000010010 |
| 11 | 000000100001101110101011110001 | 000010100011010011100000000000 | 000000000001001 |
| 12 | +  000010100011010011100000000000  =  000011000101000010001011110001 | 000101000110100111000000000000 | 000000000000100 |
| 13 | 000011000101000010001011110001 | 001010001101001110000000000000 | 000000000000010 |
| 14 | 000011000101000010001011110001 | 010100011010011100000000000000 | 000000000000001 |
| 15 | **+**  **010100011010011100000000000000**  **=**  **010111011111011110001011110001** | 101000110100111000000000000000 | 000000000000000 |

**2.2.5 Функціональна схема з відображенням управляючих сигналів**

***x2***

***x1***

***W2***

***ShR***

***DR=’0’***

***2n-1***

***0***

RG1

RG2

***0***

***n-1***

***R***

***W1***

***n***

***2n***

***X***

***2n-1***

***2n-1***

***0***

***0***

SM

***2n***

***2n***

***W3***

***ShL***

***DL=’0’***

RG3

***0***

***2n-1***

***n***

***n***

***0***

***Y***

*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

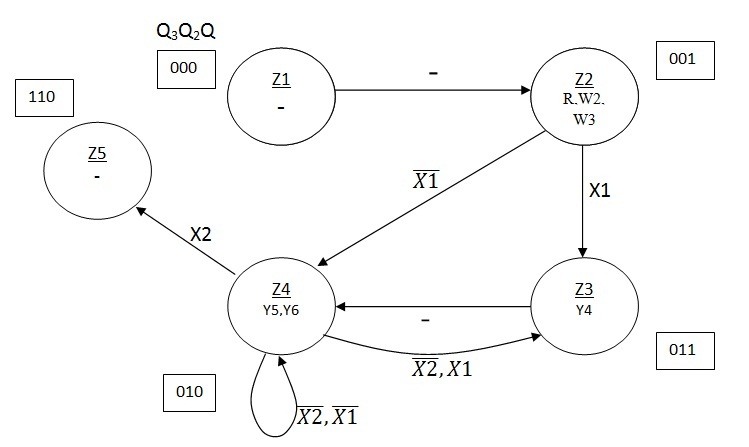
Z3

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 і нормалізація**

.

Отримали результат: Mz = 0,010111011111011 <=; ;

Mz = 0,101110111110110, ; .

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 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 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

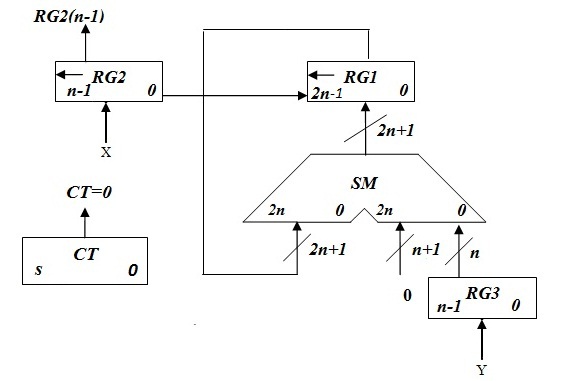
*.*

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

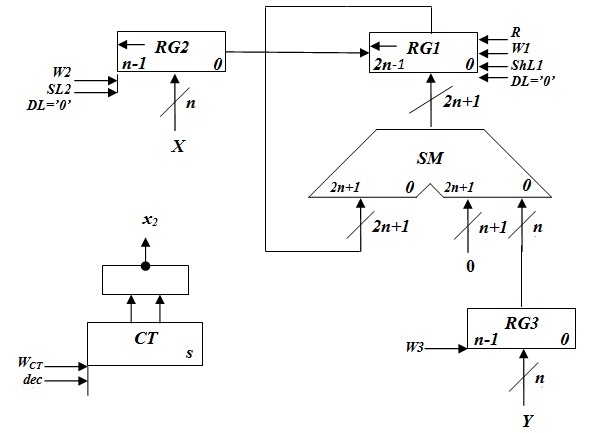
*Рисунок 2.3.2. Змістовний мікроалгоритм пристрою множення третім способом*

**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 | 101000110100111 | 100100110100111 | 1111 |
| 1 | 000000000000001010001101001110 |  | 001001101001110 | 1110 |
| 2 | 000000000000010100011010011100 |  | 010011010011100 | 1101 |
| 3 | 000000000000101000110100111000 |  | 100110100111000 | 1100 |
| 4 | +  000000000000000101000110100111  =  000000000000101101111011011111  000000000001011011110110111110 |  | 001101001110000 | 1011 |
| 5 | 000000000010110111101101111100 |  | 011010011100000 | 1010 |
| 6 | 000000000101101111011011111000 |  | 110100111000000 | 1001 |
| 7 | +  000000000000000101000110100111  =  000000000101110100100010011111  000000001011101001000100111110 |  | 101001110000000 | 1000 |
| 8 | +  000000000000000101000110100111  =  000000000101110100100010011111  000000001011101001000100111110 |  | 010011100000000 | 0111 |
| 9 | 000000101110111000101110010100 |  | 100111000000000 | 0110 |
| 10 | +  000000000000000101000110100111  =  000000101110111101110100111011  000001011101111011101001110110 |  | 001110000000000 | 0101 |
| 11 | 000010111011110111010011101100 |  | 011100000000000 | 0100 |
| 12 | 000101110111101110100111011000 |  | 111000000000000 | 0011 |
| 13 | +  000000000000000101000110100111  =  000101110111110011101101111111  001011101111100111011011111110 |  | 110000000000000 | 0010 |
| 14 | +  000000000000000101000110100111  =  001011101111101100100010100101  010111011111011001000101001010 |  | 100000000000000 | 0001 |
| 15 | +  000000000000000101000110100111  =  010111011111011110001011110001 |  | 000000000000000 | 0000 |

2.3.5 **Функціональна схема з відображенням управляючих сигналів**

**

*Рисунок 2.3.3 Функціональна схема пристрою множення третім способом*

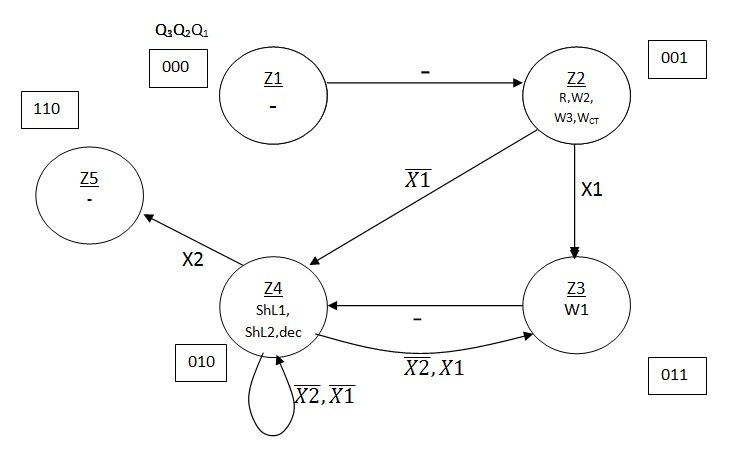
2.3.6 **Закодований мікроалгоритм**

*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 і нормалізація**

.

Отримали результат: Mz = 0,010111011111011 <=; ;

Mz = 0,101110111110110, ; .

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

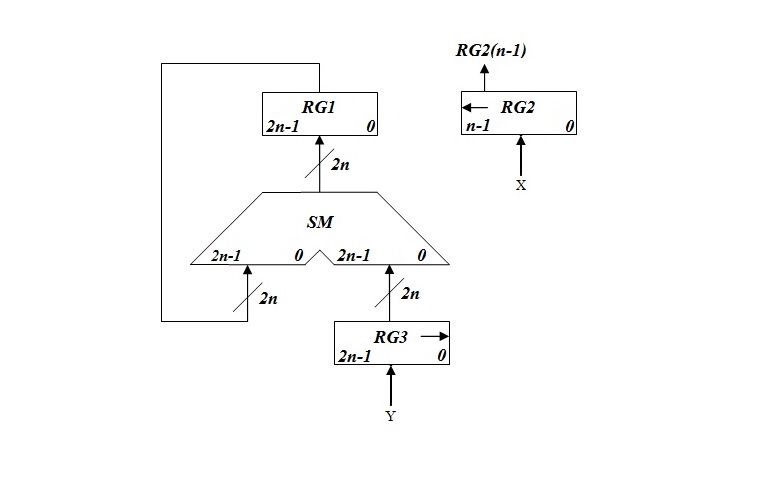
*.*

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

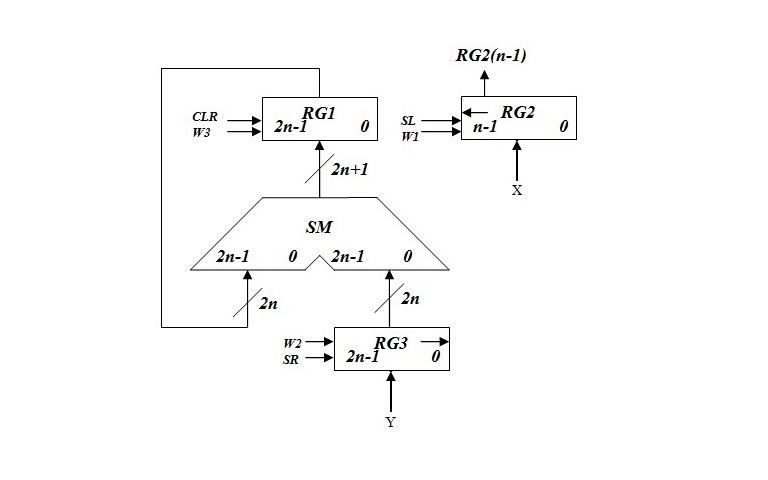
*Рисунок 2.4.2. Змістовний мікроалгоритм пристрою множення четвертим способом*

**2.4.4** **Таблиця станів регістрів**

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

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG3 | RG2 |
| П.С. | 000000000000000000000000000000 | 101000110100111000000000000000  010100011010011100000000000000 | 101000110100111 |
| 1 | 010100011010011100000000000000 | 001010001101001110000000000000 | 010001101001110 |
| 2 | 010100011010011100000000000000 | 000101000110100111000000000000 | 100011010011100 |
| 3 | 010100011010011100000000000000 | 000010100011010011100000000000 | 000110100111000 |
| 4 | +  000010100011010011100000000000  =  010110111101101111100000000000 | 000001010001101001110000000000 | 001101001110000 |
| 5 | 010110111101101111100000000000 | 000000101000110100111000000000 | 011010011100000 |
| 6 | 010110111101101111100000000000 | 000000010100011010011100000000 | 110100111000000 |
| 7 | +  000000010100011010011100000000  =  010111010010001001111100000000 | 000000001010001101001110000000 | 101001110000000 |
| 8 | +  000000001010001101001110000000  =  010111011100010111001010000000 | 000000000101000110100111000000 | 010011100000000 |
| 9 | 010111011100010111001010000000 | 000000000010100011010011100000 | 100111000000000 |
| 10 | +  000000000010100011010011100000  =  010111011110111010011101100000 | 000000000001010001101001110000 | 001110000000000 |
| 11 | 010111011110111010011101100000 | 000000000000101000110100111000 | 011100000000000 |
| 12 | 010111011110111010011101100000 | 000000000000010100011010011100 | 111000000000000 |
| 13 | +  000000000000010100011010011100  =  010111011111001110110111111100 | 000000000000001010001101001110 | 110000000000000 |
| 14 | +  000000000000001010001101001110  =  010111011111011001000101001010 | 000000000000000101000110100111 | 100000000000000 |
| 15 | +  000000000000000101000110100111  =  010111011111011110001011110001 | 000000000000000010100011010011 | 000000000000000 |

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

**

*Рисунок 2.4.3. Функціональна схема пристрою множення четвертим способом*

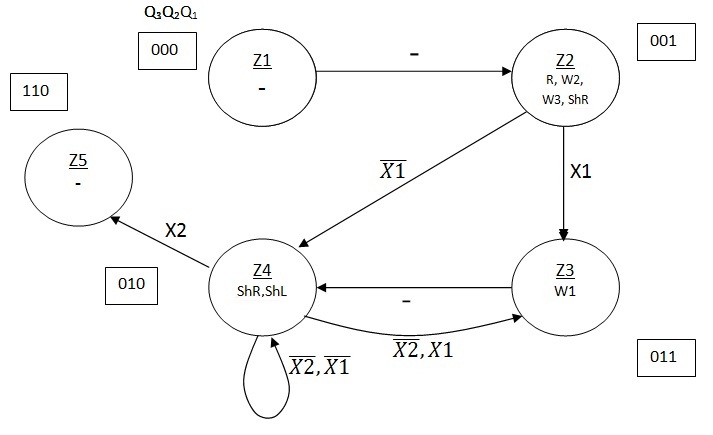
2.4.6 **Закодований мікроалгоритм**

*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*

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

**

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

**2.4.8 Обробка порядків і нормалізація**

.

Отримали результат: Mz = 0,010111011111011 <=; ;

Mz = 0,101110111110110, ; .

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 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |

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

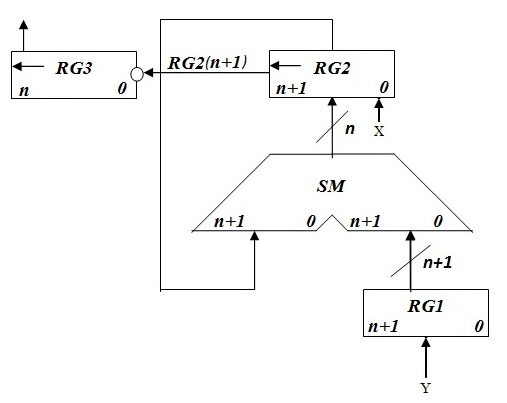
**2.5.1** **Теоретичне обґрунтування способу**

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;

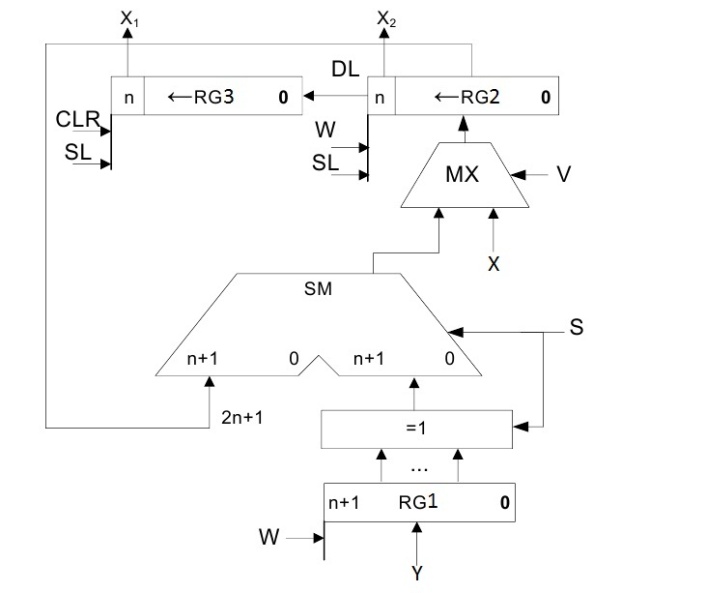
*Рисунок 2.5.2. Змістовний мікроалгоритм пристрою ділення першим способом*

**2.5.4** **Таблиця станів регістрів**

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

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 |
| П.С. | 0000000000000000 | 00100100110100111 | 00101000110100111пк  11010111001011001дк |
| 1 | 000000000000000**1** | 01001001101001110  +  11010111001011001  =  00100000110100111 |  |
| 2 | 00000000000000**1**1 | 01000001101001110  +  11010111001011001  =  00011000110100111 |  |
| 3 | 0000000000000**1**11 | 00110001101001110  +  11010111001011001  =  00001000110100111 |  |
| 4 | 000000000000**1**111 | 00010001101001110  +  11010111001011001  =  11101000110100111 |  |
| 5 | 00000000000**1**1110 | 11010001101001110  +  00101000110100111  =  11111010011110101 |  |
| 6 | 0000000000**1**11100 | 11110100111101010  +  00101000110100111  =  00011101110010001 |  |
| 7 | 000000000**1**111001 | 00111011100100010  +  11010111001011001  =  00010010101111011 |  |
| 8 | 00000000**1**1110011 | 00100101011110110  +  11010111001011001  =  11111100101001111 |  |
| 9 | 0000000**1**11100110 | 11111001010011110  +  00101000110100111  =  00100010001000101 |  |
| 10 | 000000**1**111001101 | 01000100010001010  +  11010111001011001  =  00011011011100011 |  |
| 11 | 00000**1**1110011011 | 00110110111000110  +  11010111001011001  =  00001110000011111 |  |
| 12 | 0000**1**11100110111 | 00011100000111110  +  11010111001011001  =  11110011010010111 |  |
| 13 | 000**1**111001101110 | 11100110100101110  +  00101000110100111  =  00001111011010101 |  |
| 14 | 00**1**1110011011101 | 00011110110101010  +  11010111001011001  =  11110110000000011 |  |
| 15 | 0**1**11100110111010 | 11101100000000110  +  00101000110100111  =  00010100110101101 |  |
| 16 | **1,111001101110101** | 00101001101011010  +  11010111001011001  =  00000000110110011 |  |

**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

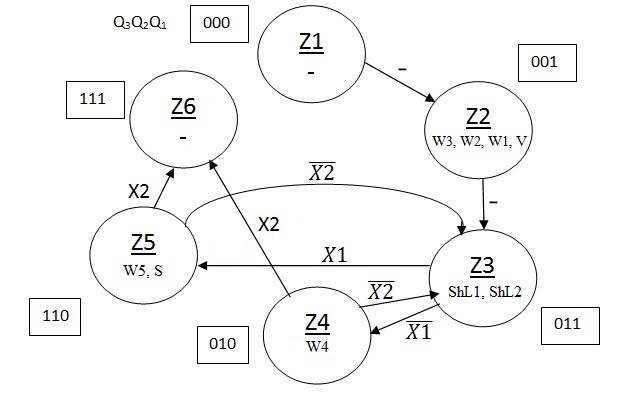
Z4

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= ,111100110111010

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 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |

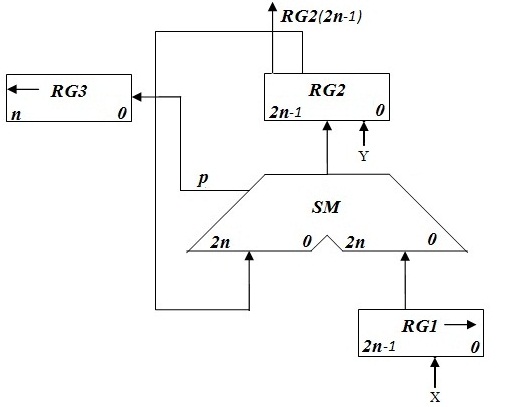
**2.6** **Другий спосіб ділення (без відновлення остачі)**

**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

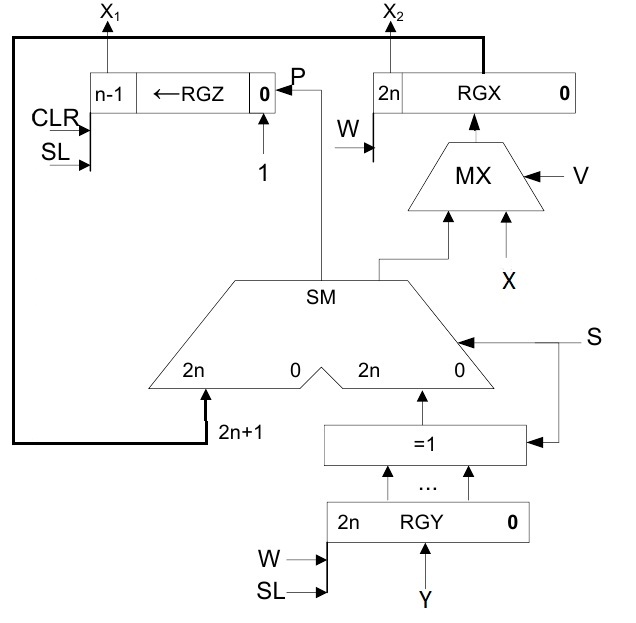
*Рисунок 2.6.2. Змістовний мікроалгоритм пристрою ділення другим способом*

**2.6.4** **Таблиця станів регістрів**

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

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 |
| П.С. | 000000000000000**1** | 010010011010011100000000000000 | 001010001101001110000000000000 |
| 1 | 00000000000000**1**1 | +  110101110010110010000000000000  =  001000001101001110000000000000 | 000101000110100111000000000000 |
| 2 | 0000000000000**1**11 | +  111010111001011001000000000000  =  000011000110100111000000000000 | 000010100011010011100000000000 |
| 3 | 000000000000**1**111 | +  111101011100101100100000000000  =  000000100011010011100000000000 | 000001010001101001110000000000 |
| 4 | 00000000000**1**1110 | +  111110101110010110010000000000  =  111111010001101001110000000000 | 000000101000110100111000000000 |
| 5 | 0000000000**1**11100 | +  000000101000110100111000000000  =  111111111010011110101000000000 | 000000010100011010011100000000 |
| 6 | 000000000**1**111001 | +  000000010100011010011100000000  =  000000001110111001000100000000 | 000000001010001101001110000000 |
| 7 | 00000000**1**1110011 | +  111111110101110010110010000000  =  000000000100101011110110000000 | 000000000101000110100111000000 |
| 8 | 0000000**1**11100110 | +  111111111010111001011001000000  =  111111111111100101001111000000 | 000000000010100011010011100000 |
| 9 | 000000**1**111001101 | +  000000000010100011010011100000  =  000000000010001000100010100000 | 000000000001010001101001110000 |
| 10 | 00000**1**1110011011 | +  111111111110101110010110010000  =  000000000000110110111000110000 | 000000000000101000110100111000 |
| 11 | 0000**1**11100110111 | +  111111111111010111001011001000  =  000000000000001110000011111000 | 000000000000010100011010011100 |
| 12 | 000**1**111001101110 | +  111111111111101011100101100100  =  111111111111111001101001011100 | 000000000000001010001101001110 |
| 13 | 00**1**1110011011101 | +  000000000000001010001101001110  =  000000000000000011110110101010 | 000000000000000101000110100111 |
| 14 | 0**1**11100110111010 | +  111111111111111010111001011001  =  111111111111111110110000000011 | 000000000000000010100011010011 |
| 15 | **1,111001101110101** | +  000000000000000010100011010011  =  000000000000000001010011010110 | 000000000000000001010001101001 |

**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

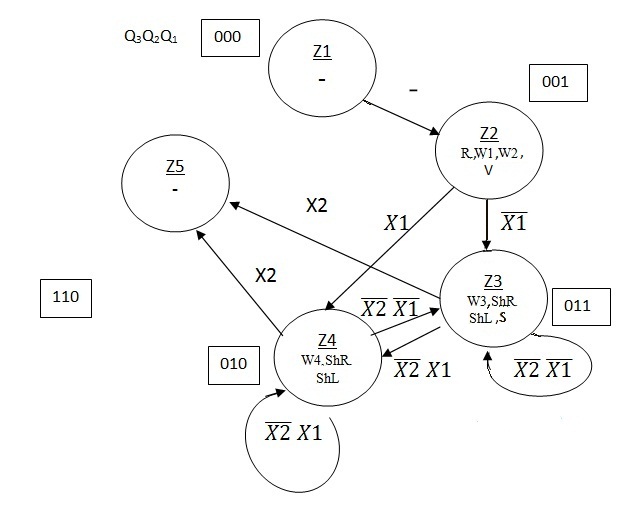
Z3

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= ,111100110111010

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 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 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 |
| 0.101000110100111 | 11 | П.С. |
| 0.010100011010011 | 10 | 🡪 |
| 0.001010001101001 | 01 | 🡪 |
| 0.000101000110100 | 00 | 🡪 |

**Додавання мантис у модифікованому ДК**

*Таблиця 2.7.2. Додавання мантис*

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

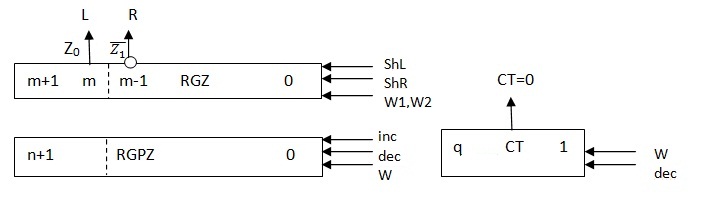
*Рисунок 2.7.2. Змістовний мікроалгоритм пристрою додавання двох чисел*

**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 |  | 0 | 1 | 100 |  |
| **1** | 000111 |  | 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**

**

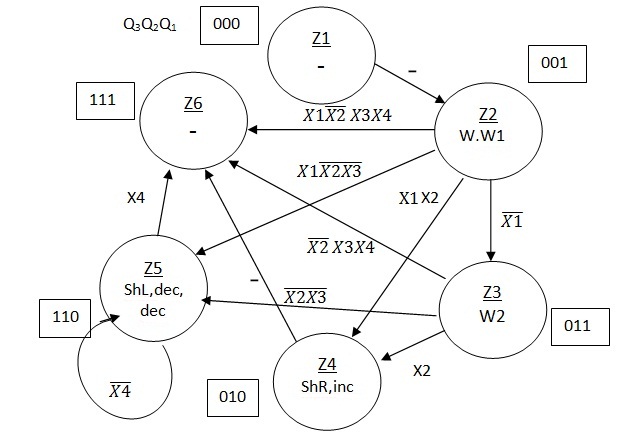
*Рисунок 2.7.4. Функціональна схема пристрою додавання двох чисел*

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

**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.

;

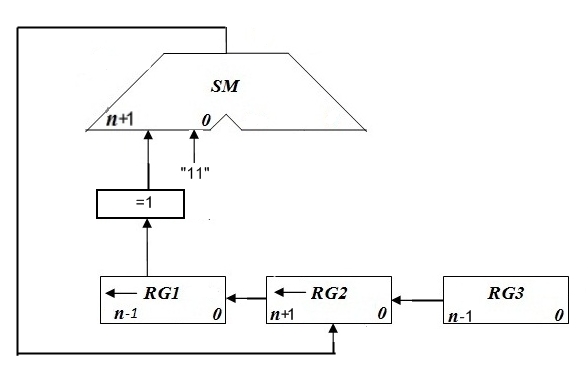
2. Якщо , то .

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

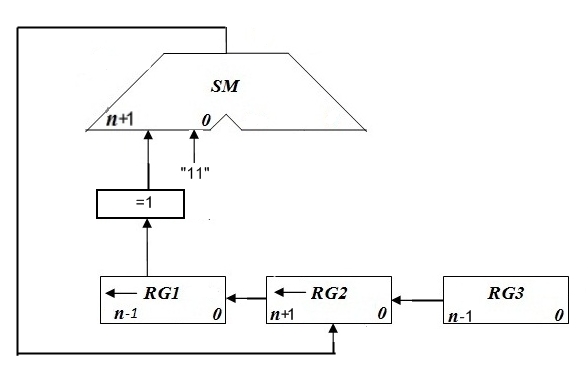
*Рисунок 2.8.2. Змістовний мікроалгоритм пристрою добування кореня числа*

**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  00000000000000010 | 101000110100111 | 1111 |
| 1 | 000000000000001 | +11111111111111111  =  00000000000000001  00000000000000110 | 010100011010011 | 1110 |
| 2 | 000000000000011 | +11111111111111011  =  00000000000000001  00000000000000100 | 001010001101001 | 1101 |
| 3 | 000000000000110 | +11111111111110011  =  11111111111110111  11111111111011111 | 000101000110100 | 1100 |
| 4 | 000000000001100 | +00000000000011011  =  11111111111111010  11111111111101001 | 000010100011010 | 1011 |
| 5 | 000000000011001 | +00000000000110011  =  00000000000011100  00000000001110000 | 000001010001101 | 1010 |
| 6 | 000000000110011 | +11111111110011011  =  00000000000001011  00000000000101111 | 000000101000110 | 1001 |
| 7 | 000000001100110 | +11111111100110011  =  11111111101100010  11111110110001010 | 000000010100011 | 1000 |
| 8 | 000000011001100 | +00000000110011011  =  11111111100100101  11111110010010100 | 000000001010001 | 0111 |
| 9 | 000000110011000 | +00000001100110011  =  11111111111000111  11111111100011100 | 000000000101000 | 0110 |
| 10 | 000001100110001 | +00000011001100011  =  00000010101111111  00001010111111100 | 000000000010100 | 0101 |
| 11 | 000011001100011 | +11111001100111011  =  00000100100110111  00010010011011100 | 000000000001010 | 0100 |
| 12 | 000110011000111 | +11110011001110011  =  00000101101001111  00010110100111100 | 000000000000101 | 0011 |
| 13 | 001100110001110 | +11100110011100011  =  11111101000011111  11110100001111100 | 000000000000010 | 0010 |
| 14 | 011001100011101 | +00110011000111011  =  00100111010110111  10011101011011100 | 000000000000001 | 0001 |
| 15 | **110011000111011** | +01100110001110111  =  00000011101010011  00001110101001100 | 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

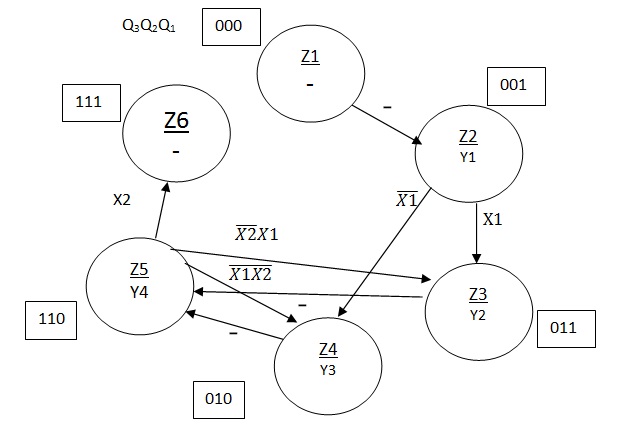
Z3

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 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |

**3**.**Синтез управляючого автомату Мура**

x3x2x1+1=1002=410 – операція множення четвертим 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 четвертим способом*

|  |  |
| --- | --- |
| Входи операційного автомата | Виходи управляючого автомата |
| R,W2,W3,ShR | Y1 |
| W1 | Y2 |
| ShR,ShL | 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

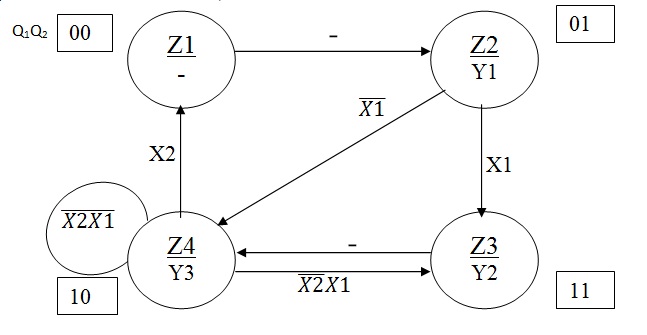
Z3

Z4

Z1

*Рисунок 3.1. Закодований мікроалгоритм пристрою множення четвертим способом*

Будуємо граф автомата Мура



*Рисунок 3.2. Граф автомата Мура пристрою множення четвертим 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.

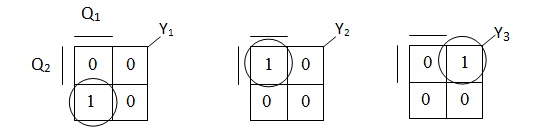
*Таблиця 3.2. Структурна таблиця автомата пристрою множення третім способом*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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



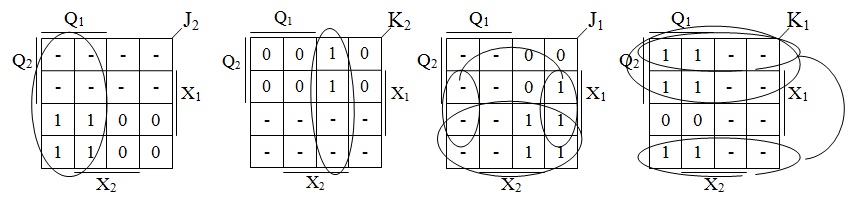
Q1

Q2

Q2

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 четвертим 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 четвертим способом та побудував функціональну схему цього автомату.