НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра обчислювальної техніки

РОЗРАХУНКОВА ГРАФІЧНА РОБОТА

з дисципліні "Комп’ютерна логіка 2. Комп’ютерна арифметика "

Виконав

Бурбіль Максим Андрійович

Факультет ІОТ,

Група ІО - 62

Залікова книжка № 6203

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

(підпис керівника)

Київ – 2017 р.

**Завдання:**

1. Числа  і  в прямому коді записати у формі з плаваючою комою у класичному варіанті (з незміщеним порядком і повною мантисою). На порядок відвести 4 розряди, на мантису 7 розрядів (з урахуванням знакових розрядів). Записати числа і  також за стандартом ANSI/IEEE 754-2008 в короткому 32-розрядному форматі).

2. Виконати 8 операцій з числами, що подані з плаваючою комою в класичному варіанті (чотири способи множення, два способи ділення, додавання та обчислення кореня додатного числа). Номери операцій (для п.3) відповідають порядку переліку, починаючи з нуля (наприклад, 0 – множення першим способом; 5 – ділення другим способом). Операндами для першого способу множення є задані числа та . Для кожної наступної операції першим операндом є результат попередньої операції, а другим операндом завжди є число . (Наприклад, для ділення першим способом першим операндом є результат множення за четвертим способом, для операції обчислення кореня операндом є результат додавання зі знаком плюс).

Для обробки мантис кожної операції, подати:

2.1 теоретичне обґрунтування способу;

2.2 операційну схему;

2.3 змістовний (функціональний) мікроалгоритм;

2.4 таблицю станів регістрів (лічильника), довжина яких забезпечує одержання 6 основних розрядів мантиси результату;

2.5 обробку порядків (показати у довільній формі);

2.6 форму запису нормалізованого результату з плаваючою комою в пам’ять комп’ютера в прямому коді.

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

3 Для операції з номером  додатково виконати:

3.1 побудувати функціональну схему з відображенням управляючих сигналів, входів для запису операндів при ініціалізації пристрою і схем формування внутрішніх логічних умов;

3.2 розробити закодований (структурний) мікроалгоритм (мікрооперації замінюються управляючими сигналами виду W,SL,SR тощо);

3.3 для операції з парним двійковим номером  додатково подати граф управляючого автомата Мура з кодами вершин, а для непарного номера – автомата Мілі;

3.4 побудувати управляючий автомат на тригерах та елементах булевого базису. Вибрати -тригери для автомата Мура та -тригери для автомата Мілі.

**Об**ґ**рунтування варіанту**

620310=1 1000 0011 10112

X = – 011110, 0111

Y­ = +010011, 1011

**Ocновна частина**

**Завдання №1**

X ПК = 1.011110, 0111

Y­ПК = 0.010011, 1011

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

X:

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

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

Y:

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

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

**Представлення чисел у формі з плаваючою точкою з характеристикою і мантисою:**

**E = P + 2m ;**

**m = 7;**

**27 = 100000002;**

**Ex = 10000000 + 101 = 10000101;**

**Х:**

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

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

**Ey = 10000000 + 101 = 10000101;**

**Y:**

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

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

**Представлення за стандартом ANSI/IEEE 754-2008**

X = -0, 0111100111

**X в IEEE 754***:*

S = 1

Експонента =5 + 128 = 13310 = 100001012

Мантиса = 0110001112

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S | Експонента | | | | | | | | Мантиса | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Y = +0, 0100111011

**Y в IEEE 754:**

S = 0

Експонента *= 5 + 128 = 13310 = 100001012*

Мантиса = 010100112

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S | Експонента | | | | | | | | Мантиса | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**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:

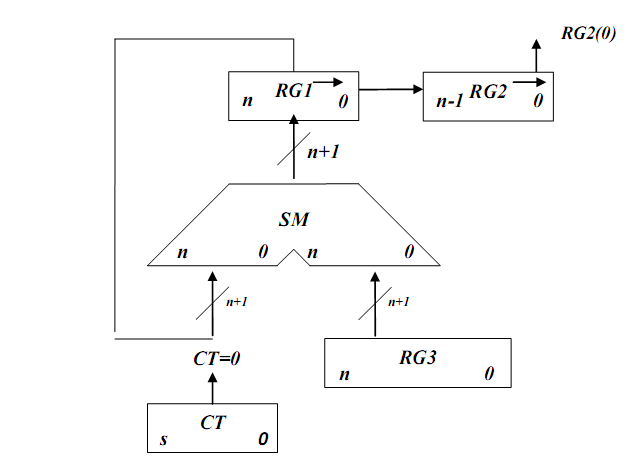
Z = YХ = Y*хn*2*-n*+Y*хn-1*2*-n+1­+*…+ Y*х*12-1;



Z = YХ = ((…((0+Y*х*n)2-1+ Y*х*n-1)2-1 +…+ Y*х*1)2-1;

Z = (Zi-1+Y*х*n-i+1)2-1.

**2.1.2 Операційна схема:**



*Рис. 2.1.1 - Операційна схема*

**2.1.3 Змістовний мікроалгоритм:**

Початок

RG1:=0; RG2:=X; RG3:=Y; CT:=6;

RG2(n)

RG1:=RG1+RG3;

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

CT:=CT-1;

CT = 0

Кінець

1

0

1

0

*Рис. 2.1.2 - Змістовний мікроалгоритм виконання операції множення першим способом*

**2.1.4 Таблиця станів регістрів:**

*Таблиця 2. 1. 1*

**Таблиця станів регістрів для першого способу множення**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **RG1** | **RG2** | **RG3** | **CT** |
| **ПС** | 0000000 | 111100 | 100111 | 110 |
| **1** | 0000000 | 011110 |  | 101 |
| **2** | 0000000 | 001111 |  | 100 |
| **3** | 0010011 | 100111 |  | 011 |
| **4** | +  0100111  =  0111010  0011101 | 010011 |  | 010 |
| **5** | +  0100111  =  1000100  0100010 | 001001 |  | 001 |
| **6** | +  0100111  =  1001001  **0100100** | **100100** |  | 000 |

**2.1.5 Обробка порядків:**

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків: PX + PY = PZ .



PX = 5; PY = 5; PZ = 1010 = 10102



**2.1.6 Нормалізація результату:**

Отримали результат: **0100100100100**.

Знак мантиси: 

Робимо зсув результату вліво, доки у першому розряді не буде одиниця:

**100100100100**.

Порядок зменшуємо на 1: PZ = 9.



Запишемо нормалізований результат: 100101

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

**2.2 Другий спосіб множення**

**2.2.1 Теоретичне обґрунтування другого способу множення:**

Числа множаться у прямих кодах. Під час множення чисел 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 другим способом здійснюється з молодших розрядів, множене зсувається вліво, а сума часткових добутків залишається нерухомою. Перед початком множення другим способом множник X записують у регістр RG2, а множене Y – в молодші розряди регістру RG3(тобто в регістрі RG3 установлюють Y0 = Y2-n ). В кожному i-му циклі множення додаванням кодів RG3 і RG1 керує цифра RG2(n), а в регістрі RG3 здійснюється зсув вліво на один розряд, у результаті чого формується величина Yi = 2Yi-1 . Оскільки сума часткових добутків у процесі множення нерухома, зсув у регістрі RG3 можна сполучити в часі з підсумовуванням (як правило, tП>tЗ). Завершення операції множення визначається за нульовим вмістом регістру RG2, що також приводить до збільшення швидкодії, якщо множник ненормалізований.

Вираз: Z = YХ = Y*хn*2*-n*+Y*хn-1*2*-n+1­+*…+ Y*х*12-1 подамо у вигляді



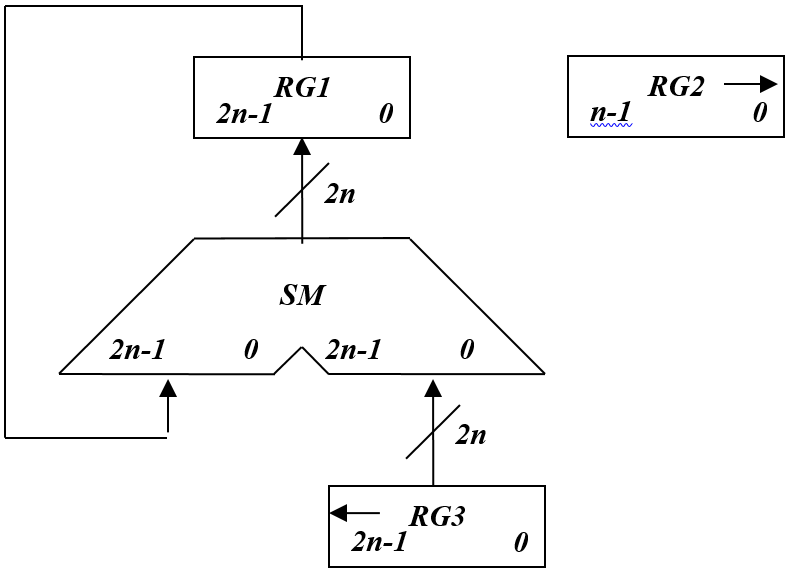
Z = ((…((0 + *Y* 2*-n хn*) + *Y* 2*-n+1 х*n-1)+…+ *Y* 2-1 *х*1.

Отже, сума часткових добутків в i-му циклі, де , зводиться до обчислення виразу: Z*i* = Z*i-1* +*Yi  хn-i+1*, Y*i* = 2Y*i-1.*



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

**2.2.2 Операційна схема:**



*Рис. 2.2.1 - Операційна схема*

**2.2.3 Змістовний мікроалгоритм:**



*Рис. 2.2.2 - Змістовний мікроалгоритм*

**2.2.4 Таблиця станів регістрів:**

*Таблиця 2.2.1*

**Таблиця станів регістрів**

|  |  |  |  |
| --- | --- | --- | --- |
| №ц. | RG1 | RG2🡪 | RG3🡨 |
| П.С. | 000000000000 | 100101 | 000000100111 |
| 1 | 000000100111 | 010010 | 000001001110 |
| 2 | 000000100111 | 001001 | 000010011100 |
| 3 | +  000010011100  =  000011000011 | 000100 | 000100111000 |
| 4 | 000011000011 | 000010 | 001001110000 |
| 5 | 000011000011 | 000001 | 010011100000 |
| 6 | +  010011100000  =  **010110100011** | 000000 | 100111000000 |

**2.2.5 Обробка порядків:**

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків: PX + PY = PZ .



PX = 9; PY = 5; PZ = 1410 = 11102.



**2.2.6 Нормалізація результату:**

Отримали результат: **010110100011**.

Знак мантиси: 

Робимо здвиг результату вліво, доки у першому розряді не буде одиниця:

**10110100011**.

Порядок зменшуємо на 1:

PZ = 13.



Запишемо нормалізований результат:

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

**2.3 Третій спосіб множення**

**2.3.1 Теоретичне обгрунтування третього способу множення:**

Числа множаться у прямих кодах. Під час множення чисел 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 третім способом здійснюється зі старших розрядів множника, сума часткових добутків зсувається вліво, а множене нерухоме. Під час множення третім способом вага молодшого розряду RG3 дорівнює 2-2n , тому код у регістрі RG3 являє собою значення Y2-n. На початку кожного циклу множення здійснюється лівий зсув у регістрах RG1 і RG2, а потім виконується додавання, яким керує RG1(1). У результаті підсумовування вмісту RG3 і RG1 може виникнути перенос у молодший розряд регістру RG1. У старшій частині суматора, на якому здійснюється підсумовування коду RG2 з нулями, відбувається поширення переносу. Збільшення довжини RG2 на один розряд усуває можливість поширення переносу в розряди множника. Після виконання n циклів молодші розряди добутку будуть знаходитися в регістрі RG2, а старші – в регістрі RG1. Час множення третім способом визначається аналогічно першому способу і дорівнює tm= n(tП + tЗ).

Вираз Z = YХ = Y*хn*2*-n*+Y*хn-1*2*-n+1­+*…+ Y*х*12-1 подамо у вигляді

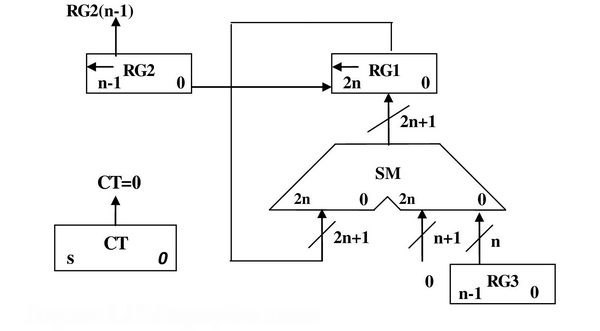


Z = ((… ((0 + *Y* 2*-n х1*)2 + *Y* 2*-n х*2)2+…+ *Y* 2*-n хi*)2 + …+Y2*-n хn*.

Отже, сума часткових добутків в *i*-му циклі, де , зводиться до обчислення виразу: Z*i* = 2Z*i-1* +*Y* 2*-n хi* ,Y*i* = 2Y*i-1* , з початковими умовами

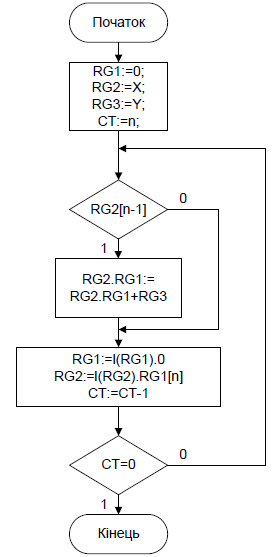
Z0 = 0, i = 1.

**2.3.2 Операційна схема:**



*Рис. 2.3.1 - Операційна схема*

**2.3.3 Змістовний мікроалгоритм:**



*Рис. 2.3.2* **-** *Змістовний мікроалгоритм*

**2.3.4 Таблиця станів регістрів:**

*Таблиця 2.3.1*

**Таблиця станів регістрів**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ ц.** | **RG1** 🡨 | **RG2** 🡨 | **RG3** | **CT** |
| **П.С.** | 000000000000 | 101101 | 100111 | 110 |
| **1** | 000001001110 | 011010 |  | 101 |
| **2** | 000010011100 | 110100 |  | 100 |
| **3** | +  000000100111  =  000011000011  000110000110 | 101000 |  | 011 |
| **4** | +  000000100111  =  000110101101  001101011010 | 010000 |  | 010 |
| **5** | 011010110100 | 100000 |  | 001 |
| **6** | +  000000100111  =  011011011011  **110110110110** | 000000 |  | 000 |

**2.3.5 Обробка порядків:**

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків: PX + PY = PZ .



PX = 13; PY = 5; PZ = 1810 = 100102.



**2.3.6 Нормалізація результату:**

Отримали результат: **110110110110**.

Знак мантиси: 

Корекція не потрібна:

**110110110110**.

PZ = 18.



Запишемо нормалізований результат:

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

**2.4 Четвертий спосіб множення**

**2.4.1 Теоретичне обґрунтування четвертого способу множення:**

Числа множаться у прямих кодах. Під час множення чисел 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 виконується зі старших розрядів множника, сума часткових добутків залишається нерухомою, а множене зсувається вправо. Перед множенням четвертим способом множник записують у регістр RG2, а множене – в старші розряди регістру RG3(тобто в RG3 установлюють Y0 = Y2-1). У кожному циклі цифра RG2(1), що знаходиться в старшому розряді регістру RG2, керує підсумовуванням, а в RG3 здійснюється правий зсув на один розряд, що еквівалентно множенню вмісту цього регістра на 2-1. Час виконання множення четвертим способом складає tM=ntП, визначається аналогічно до другого способу. Запишу четвертий спосіб в аналітичні формі. Вираз

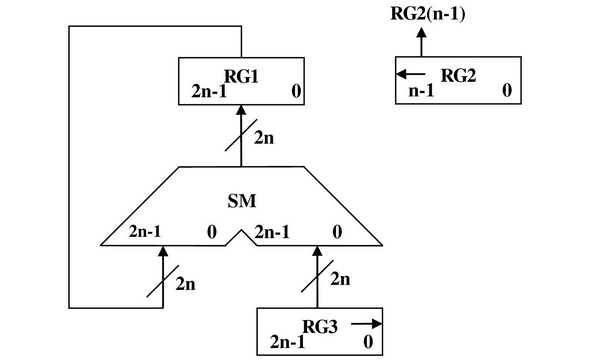
подамо у вигляді



Отже, сума часткових добутків в i-му циклі, де , зводиться до обчислення виразу: , з початковими значеннями i=1, Y0=2-1Y, Z0=0.



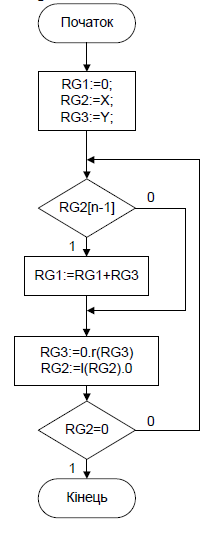
**2.4.2 Операційна схема:**



*Рисунок 2.4.1-* *Операційна схема*

**2.4.3 Змістовний мікроалгоритм:**

*Рис. 2.4.2* **-** *Змістовний мікроалгоритм*



**2.4.4 Таблиця станів регістрів:**

*Таблиця 2.4.1*

**Таблиця станів регістрів**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ ц.** | **RG1** | **RG2🡨** | **RG3 🡪** |
| **П.С.** | 000000000000 | 10110 | 010011100000 |
| **1** | 010011100000 | 101110 | 001001110000 |
| **2** | +  001001110000  =  011101010000 | 011100 | 000100111000 |
| **3** | 011101010000 | 111000 | 000010011100 |
| **4** | +  000010011100  =  011111101100 | 110000 | 000001001110 |
| **5** | +  000001001110  =  100000111010 | 100000 | 000000100111 |
| **6** | +  000000100111  =  **100001100001** | 000000 | 000000010011 |

**2.4.5 Обробка порядків:**

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків: PX + PY = PZ .



PX = 18; PY = 5; PZ = 2310 = 101112.



**2.4.6 Нормалізація результату:**

Отримали результат: **100001100001**.

Знак мантиси: 

Корекція не потрібна:

**100001100001**;

PZ = 23.



Запишемо нормалізований результат:

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

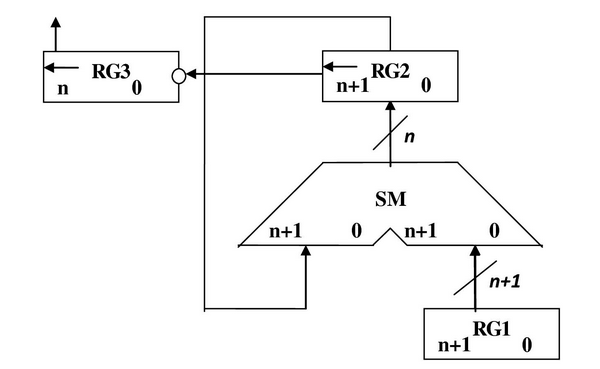
**2.5. Перший спосіб ділення**

**2.5.1 Теоретичне обґрунтування першого способу ділення:**

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. Такий спосіб називається діленням із зсувом залишку. Черговий залишок формується в регістрі RG2 (у вихідному стані в цьому регістрі записаний Х), дільник Y знаходиться в регістрі RG1. Виходи RG2 підключені до входів SM безпосередньо, тобто ланцюги видачі коду з RG2 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. Результат формується в регістрі RG3.

**2.5.2 Операційна схема:**



*Рис. 2.5.1 -* *Операційна схема*

**2.5.3 Змістовний мікроалгоритм:**

Початок

RG3:= 0

RG2:= 00.X

RG1:= 00.Y

RG2 [n+1]

Кінець

RG3:=l(RG3).

RG2:=l(RG2).0

RG2:=RG2+RG1

RG2:=RG2++1

RG2[n+1]

*Рис. 2.5.2 - Змістовний мікроалгоритм*

**2.5.4 Таблиця станів регістрів:**

*Таблиця 2.5.1*

**Таблиця станів регістрів**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ ц.** | **RG3(Z)** | **RG2(X)** | **RG1(Y)** |
| **П.С.** | 000000 | 00100010 | 00100111 |
| **1** | 0000001 | 01000100  +  11011001  =  00011101 |  |
| **2** | 0000011 | 00111010  +  11011001  =  00010011 |  |
| **3** | 0000111 | 00100110  +  11011001  =  11111111 |  |
| **4** | 0001110 | 11111110  +  00100111  =  00100101 |  |
| **5** | 0011101 | 01001010  +  11011001  =  00100011 |  |
| **6** | 0111011 | 01000110  +  11011001  =  00011111 |  |
| **7** | **1110111** | 00111110  +  11011001  =  00010111 |  |

**2.5.8 Обробка порядків:**

Порядок частки буде дорівнювати:



В моєму випадку =23; =5; =18;



**2.5.9 Нормалізація результату:**

Отримали результат: **1110111.**

Знак мантиси: 

Нормалізація мантиси не потрібна.

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

**2.6. Другий спосіб ділення**

**2.6.1 Теоретичне обґрунтування другого способу ділення:**

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**

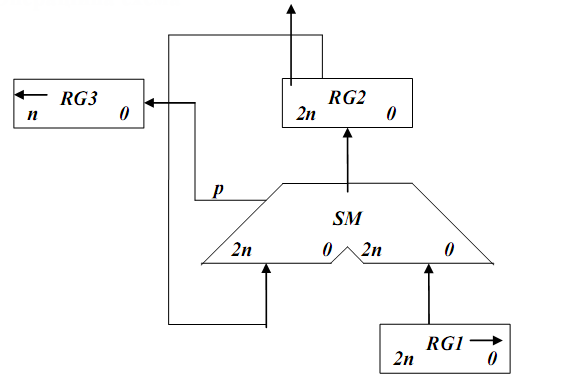
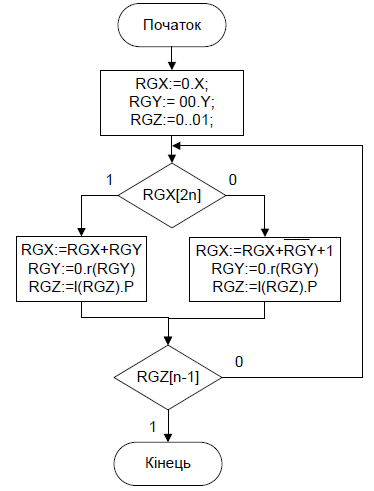


Рис. 2.6.1 - Операційнасхема для операції ділення другим способом

**2.6.3 Змістовний мікроалгоритм**



*Рис. 2.6.2 - Змістовний мікроалгоритм*

**2.6.4 Таблиця станів регістрів**

*Таблиця 2.6.1*

**Таблиця станів регістрів**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ ц.** | **RG3(Z)** | **RG2(X)** | **RG1(Y)** |
| **П.С.** | 0000001 | 011110000000 | 001001110000 |
| **1** | 0000011 | 011110000000  +  110110010000  =  010100010000 | 000100111000 |
| **2** | 0000111 | 010100010000  +  111011001000  =  001111011000 | 000010011100 |
| **3** | 0001111 | 001111011000  +  111101100100  =  001100111100 | 000001001110 |
| **4** | 0011111 | 001100111100  +  111110110010  =  001011101110 | 000000100111 |
| **5** | 0111111 | 001011101110  +  111111011001  =  001011000111 | 000000010011 |
| **6** | **1111111** | 001011000111  +  111111101101  =  001010110100 | 000000001001 |

**2.6.8 Обробка порядків:**

Порядок частки буде дорівнювати:



В моєму випадку =18; =5; =13;



**2.6.9 Нормалізація результату:**

Отримали результат: **111111.**

Знак мантиси: 

Нормалізація мантиси не потрібна.

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

**2.7. Операція додавання чисел.**

**2.7.1 Теоретичне обґрунтування способу**

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

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

1. Порівняння порядків.

Px=+1310=+11012

Py=+510=+01012



1310-510=810=10002



2. Вирівнювання порядків.

Робимо зсув вправо мантиси числа Y, зменшуючи на кожному кроці, доки не стане 0.



*Таблиця 2.7.1*

**Таблиця зсуву мантиси на етапі вирівнювання порядків**

|  |  |  |
| --- | --- | --- |
| MY | ∆ | Мікрооперація |
| 0, 100111 | 1000 | Початковий стан |
| 0, 010011 | 0111 | 🡪 |
| 0, 001001 | 0110 | 🡪 |
| 0, 000100 | 0101 | 🡪 |
| 0, 000010 | 0100 | 🡪 |
| 0, 000001 | 0011 | 🡪 |
| 0, 000000 | 0010 | 🡪 |
| 0, 000000 | 0001 | 🡪 |
| 0, 000000 | 0000 | 🡪 |

3. Додавання мантис у модифікованому ДК.

*Таблиця 2.7.2*

**Додавання мантис**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MX | 1 | 1, | 0 | 0 | 0 | 0 | 0 | 1 |
| MY | 0 | 0, | 0 | 0 | 0 | 0 | 0 | 0 |
| MZ | 1 | 1, | 0 | 0 | 0 | 0 | 0 | 1 |

MХ = 11, 111111ПК = 11, 000001ДК

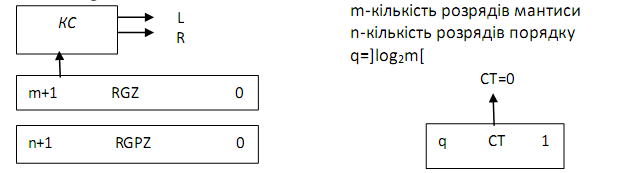
MY = 00, 000000ПК = 00, 000000ДК

MZ = 11, 000001ДК = 11, 111111ПК

4. Нормалізація результату (В ПК).

Для даного результату додавання нормалізація не потрібна.

**2.7.2 Операційна схема**



*Рис. 2.7.1-Операційна схема*

Виконаємо синтез КС для визначення порушення нормалізації.

*Таблиця 2.7.4*

**Визначення порушення нормалізації**

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



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

**2.7.3 Змістовний алгоритм**



*Рис. 2.7.2 - Змістовний мікроалгоритм*

**2.7.4 Таблиця станів регістрів**

**1) Додавання**

*Таблиця 2.7.5*

**Таблиця станів регістрів**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** **такту** | **RGPZ** | **RGZ** | **ЛПН(L)** | **ППН(R)** | **СT** | **Мікрооперація** |
| **ПС** | 001000 | 11, 111111 | 0 | 1 | 01 |  |
| **1** | 001000 | 00, 000001  00, 000000  00, 000001 | 0 | 0 | 00 |  |

**2.7.8 Обробка порядків**

PX+Y= 810 =10002

**2.7.9 Форма запису результату з плаваючою комою**

Результат додавання Z=X+Y.

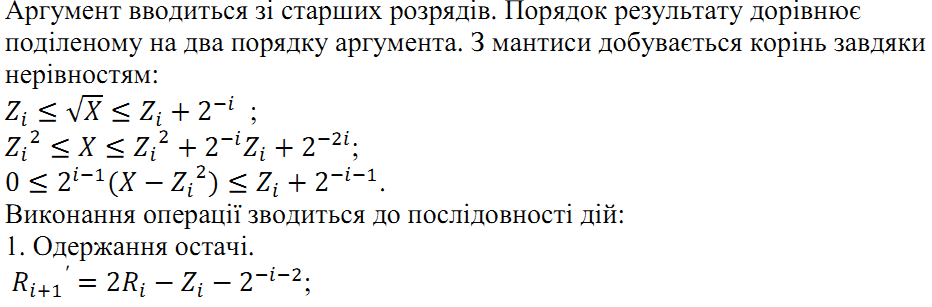
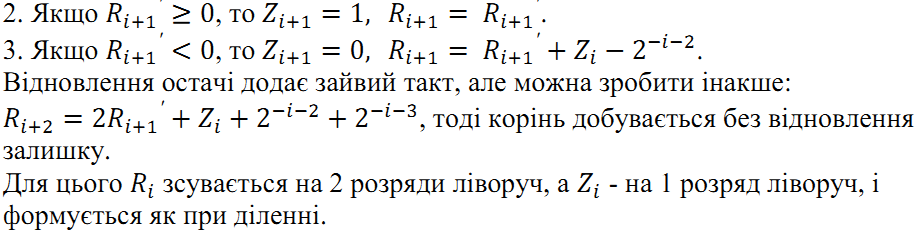
Zпк = 1. 111111ПК

Pz = 810 =10002

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

**2.8.Операція добування кореня**

**2.8.1 Теоретичне обґрунтування операції обчислення квадратного кореня**



**2.8.2 Операційна схема операції обчислення квадратного кореня**



*Рис. 2.8.1 – Операційна схема*

**2.8.3 Змістовний мікроалгоритм**



*Рис.2.8.2 – Змістовний мікроалгоритм*

**2.8.4 Таблиця станів регістрів**

*Таблиця 2.8.1*

**Таблиця станів регістрів**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | RZ | RR | RX | СТ |
| **пс** | 000000 | 00000000  00000011 | 100111 | 110 |
| **пз** |
| **1** | 000001 | 00000011  +  11111111  =  00000010  00001011 | 011111 | 101 |
| **2** | 000011 | 00001011  +  11111011  =  00000110  00011011 | 001111 | 100 |
| **3** | 000111 | 00011011  +  11110011  =  00001110  00111000 | 000111 | 011 |
| **4** | 001111 | 00111000  +  11100011  =  00011011  01101100 | 000011 | 010 |
| **5** | 011111 | 01101100  +  11000011  =  00101111  10111100 | 000001 | 001 |
| **6** | **111111** | 10111100  +  01111111  =  00111011  11101100 | 000000 | 000 |

**2.8.8 Обробка порядків**



В моєму випадку =4;



**2.8.9 Запис результату**

Отримали результат Z = 111111.

Результат нормалізований, готовий до запису у мантису:

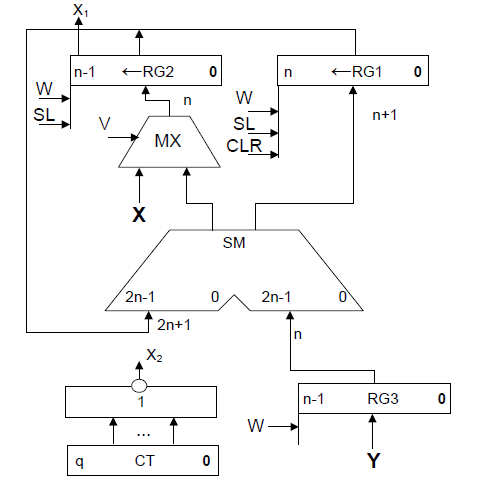
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

**Завдання №3**

Варіант завдання: четвертий спосіб множення.

Синтез управляючого автомату Мілі на RS – тригерах для операції четвертого способу множення.

**3.1 Функціональна схема:**



**3.2 Закодований мікроалгоритм:**

*Таблиця 2.3.2*

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

**Таблиця кодування сигналів**

*Таблиця 3.1*

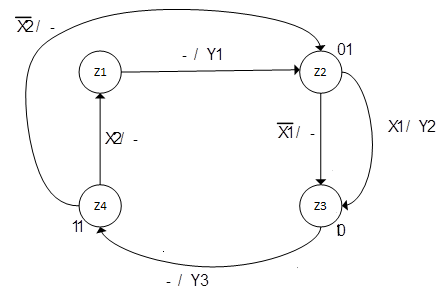
|  |  |
| --- | --- |
| W2, W3, CLR1 | Y1 |
| W1 | Y2 |
| SR3, SL2 | Y3 |

**3.3 Мікроалгоритм в термінах управляючого автомата**



*Рис. 3.1 – Закодований мікроалгоритм*

**3.3 Граф управляючого автомата**



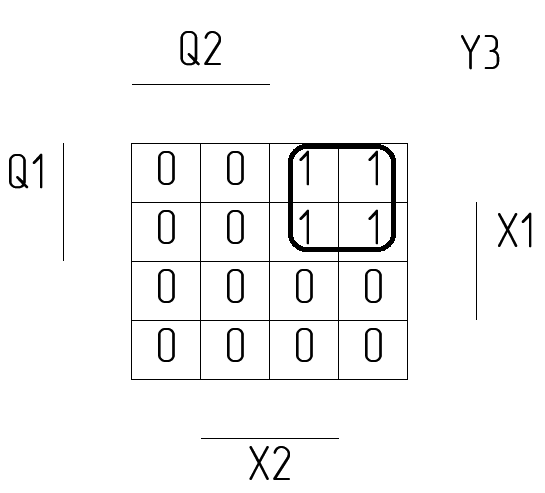
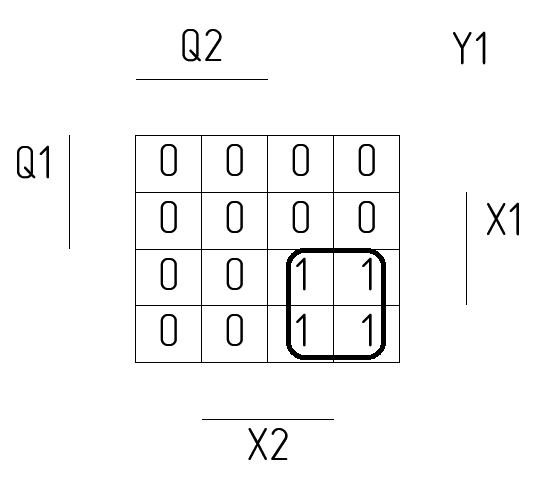
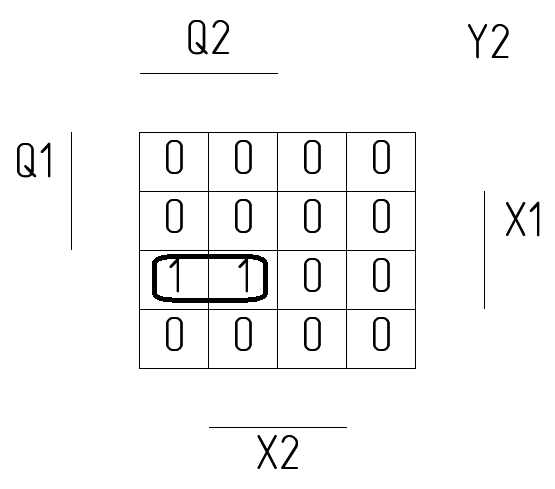
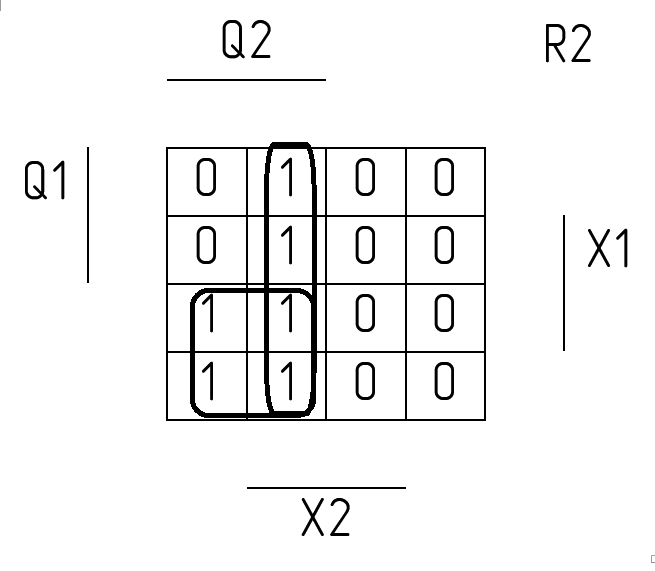
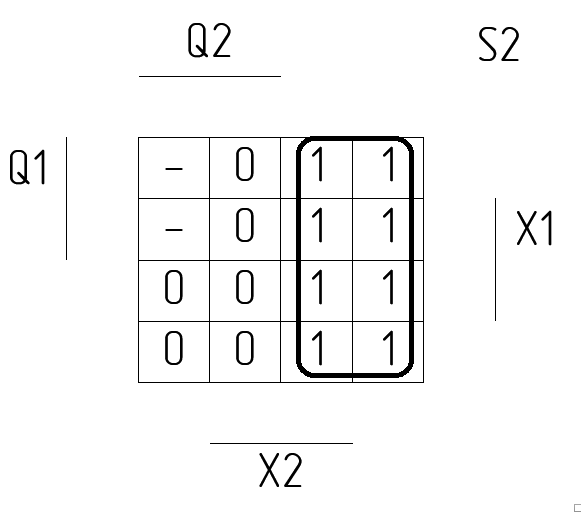
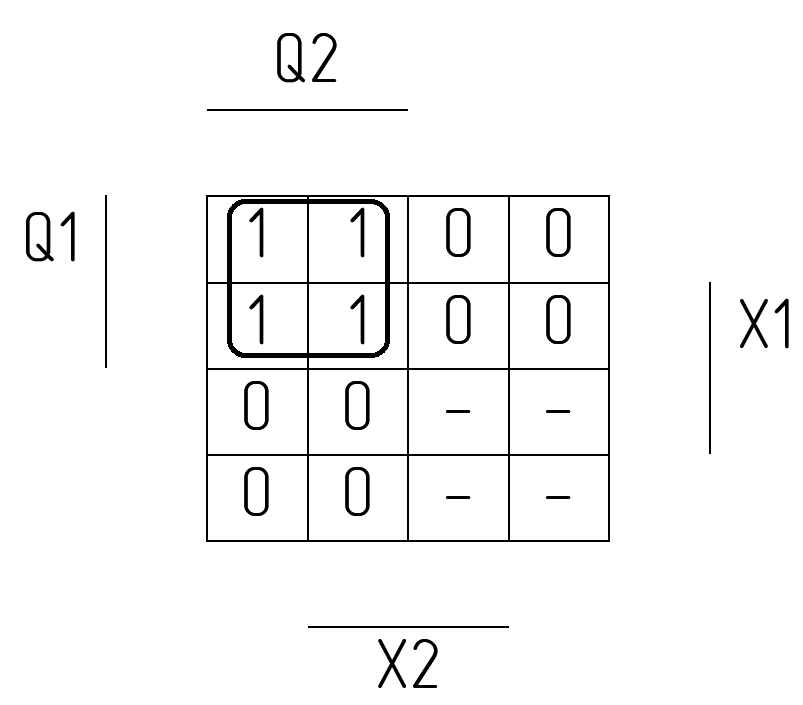
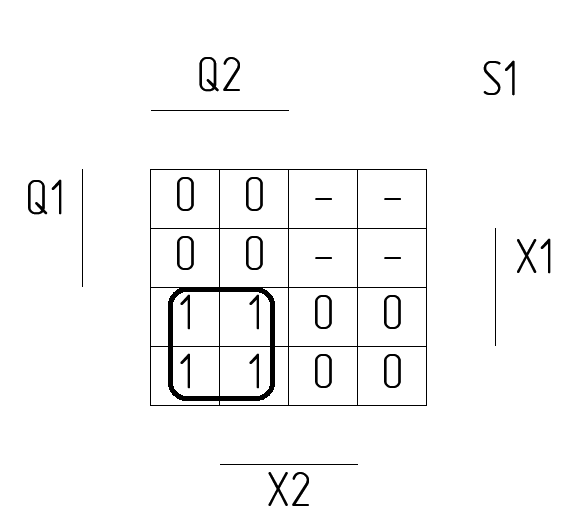
*Рис. 3.2 – Граф циклічного автомата*

**3.4 Таблиця переходів циклічного автомата на RS-тригерах**

*Таблиця 3.2*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Перехід | Старий  стан | Новий  Стан | Вхідні  сигнали | Вихідні  сигнали | Функції збудження тригерів | |
| Q1 t Q2 t | Q1 t+1 Q2 t+1 | X1 X2 | Y1 Y2 Y3 | R1 S1 | R2 S2 |
| Z1 – Z2 | 0 0 | 0 1 | - - | 1 0 0 | - 0 | 0 1 |
| Z2 – Z3 | 0 1 | 1 0 | 0 - | 0 0 0 | 0 1 | 1 0 |
| Z2 – Z3 | 0 1 | 1 0 | 1 - | 0 1 0 | 0 1 | 1 0 |
| Z3 – Z4 | 1 0 | 1 1 | - - | 0 0 1 | 0 - | 0 1 |
| Z4 – Z2 | 1 1 | 0 1 | - 0 | 0 0 0 | 1 0 | 0 - |
| Z4 – Z1 | 1 1 | 0 0 | - 1 | 0 0 0 | 1 0 | 1 0 |

**3.5 Функцій збудження тригерів. Мінімізація функцій методом діаграм Вейча :**



*Рис. 3.3 – Мінімізація функцій*

*R1 = Q1 Q2*

*R2 = Q2X2∨ Q2*



*S1 =Q2*



*S2 =*



*Y1 =*



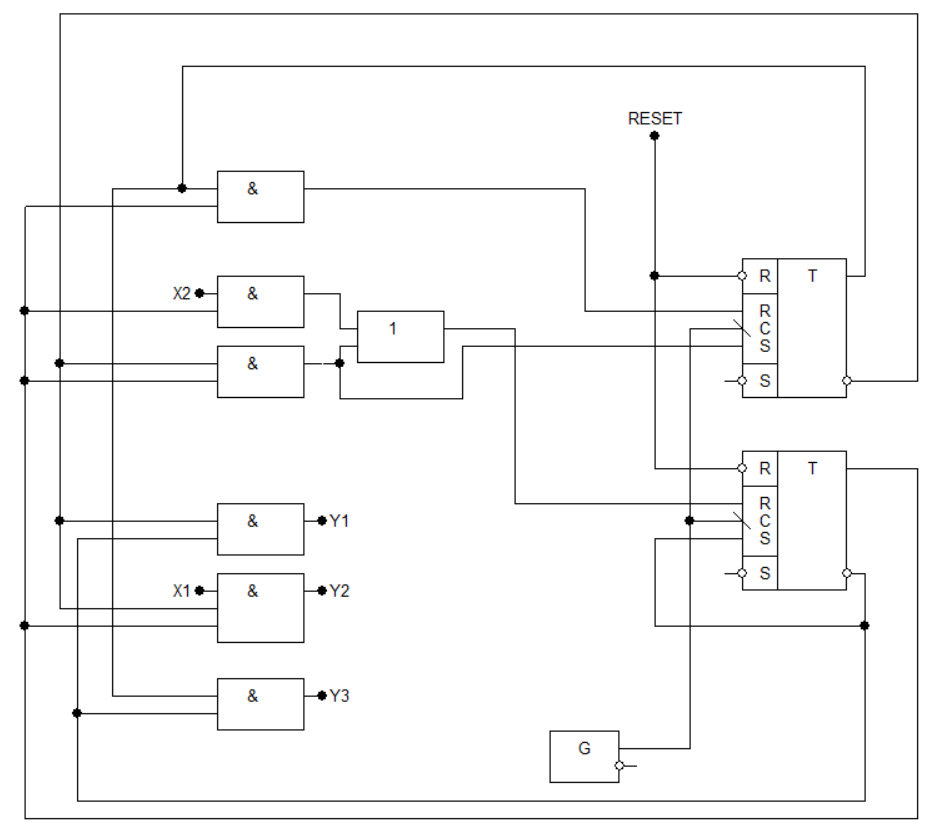
*Y2 =*



*Y3 =*



**3.6 Функціональна схема автомата**



*Рис. 3.5 - Функціональна схема*

**Висновок**

У даній розрахунковій роботі було виконано операції з числами в двійковому коді з плаваючою комою, а саме: множення чотирма способами, ділення двома способами, додавання та віднімання. Для операції ділення другим способом було побудовано управляючий автомат Мілі на RS-тригерах і елементах булевого базису. Зроблено мінімізацію функцій тригерів і в середовищі AFDK побудована функціональна схема автомата.