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

Андрійчук Дмитро Анатолійчович

Факультет ІОТ,

Група ІО-64

Залікова книжка № ІО-6401

Керівник [Верба О. А.](http://rozklad.kpi.ua/Schedules/ViewSchedule.aspx?v=3616fe25-c15f-4d3e-986b-deb3928e21b8)

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

(підпис керівника)

Київ – 2017 р.

**I. 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 першим операндом є результат додавання).

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.

**II.Обгрунтування варіанту:**

Перевести номер залікової книжки в двійкову систему. Записати два 10-розрядних двійкових числа:

 і ,

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

X = -001000,0101 Y = +110000,0001

**III.Ocновна частина:**

**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).

X = -001000,0101 Y = +110000,0001

Xпк = 1.110111,1010 Yпк = 0.110000,0001

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

Px = 0.110; Mx = 1. 1101111010;

Py = 0.110 My = 0. 1100000001

X2:

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

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

Y2:

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

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

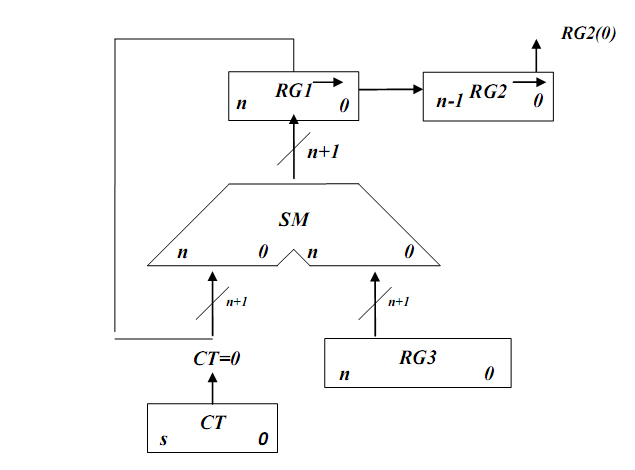
Множення мантис першим способом здійснюється з молодших розрядів множника, сума часткових добутків зсувається вправо, а множене залишається нерухомим. Тоді добуток двох чисел представляється у вигляді:

Z=Y+ Y…+ Y;

Z=(((0+Y)+ Y)…+ Y);

Z=;

**2.1.2 Операційна схема:**



*Рисунок 2.1.1- Операційна схема.*

**2.1.3 Змістовний мікроалгоритм:**

Початок

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

RG1:=RG1+RG3

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

CT=0

Кінець

1

0

1

0

RG2[0]

*Рисунок 2.1.2 - Змістовний мікроалгоритм.*

**2.1.4 Таблиця станів регістрів:**

Xпк = 1.110111 Yпк = 0.110000

*Таблиця 2.1.1-Таблиця станів регістрів.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | RG1 | RG2 | RG3 | CT |
| ПС | 0000000 | 110111 | 110000 | 110 |
| 1 | 0011000 | 011011 | 110000 | 101 |
| 2 | +  0110000  =  1001000  0100100 | 001101 | 110000 | 100 |
| 3 | +  0110000  =  1010100  0101010 | 000110 | 110000 | 011 |
| 4 | 0010101 | 000011 | 110000 | 010 |
| 4 | 0010101 | 000011 | 110000 | 010 |
| 5 | +  0110000  =  1000101  0100010 | 100001 | 110000 | 001 |
| 6 | +  0110000  =  1010010  **0101001** | **010000** | 110000 | 000 |

**2.1.5 Обробка порядків:**

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків:

=6; =6; =6+6=12=11002

**2.1.6 Нормалізація результату:**

Отримали результат:

Знак мантиси: 1 0 = 1.

Виконуємо зсув результату вліво, доки у першому розряді не опиниться одиниця, при цьому порядок числа зменшуємо на 1:

Mz = 0101001010000

101001010000← Pz=Pz-1=12-1=11=1011

Нормалізований результат:

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

**2.2 Другий спосіб множення.**

**2.2.1 Теоретичне обґрунтування другого способу множення:**

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.

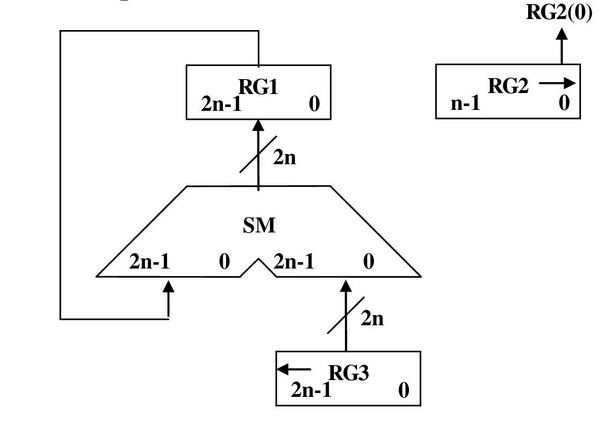
Множення мантис другим способом здійснюється з молодших розрядів, множене зсувається вліво, а сума часткових добутків залишається нерухомою.

Z=Y+ Y…+ Y;

Z=((0+ Y)+ Y)…+ Y;

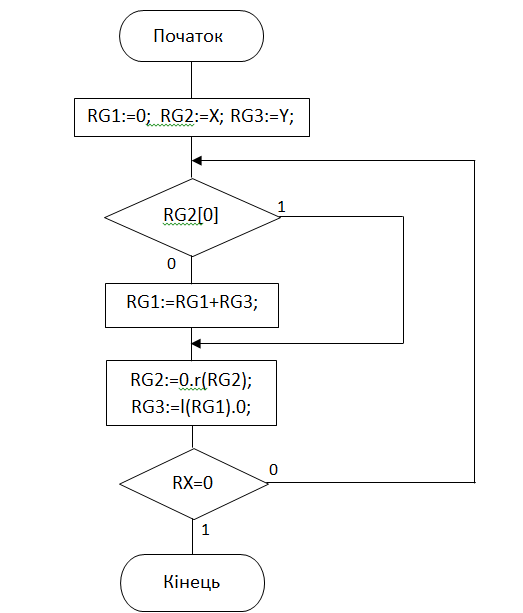
Z=;

**2.2.2 Операційна схема:**



*Рисунок 2.2.1- Операційна схема.*

**2.2.3 Змістовний мікроалгоритм:**



1

0

**2.2.4 Таблиця станів регістрів:**

Xпк = 1.101001 Yпк = 0.110000

*Таблиця 2.2.1-Таблиця станів регістрів.*

|  |  |  |  |
| --- | --- | --- | --- |
| № | RG1 | RG2 | RG3 |
| ПС | 000000000000 | 101001 | 000000110000 |
| 1 | 000000110000 | 010100 | 000001100000 |
| 2 | 000000110000 | 001010 | 000011000000 |
| 3 | 000000110000 | 000101 | 000110000000 |
| 4 | +  000110000000  =  000110110000 | 000010 | 001100000000 |
| 5 | 000110110000 | 000001 | 011000000000 |
| 6 | **+**  011000000000  =  **011110110000** |  | 110000000000 |

**2.2.5 Обробка порядків:**

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків:

=1110; =610; =1710=100012

**2.2.6 Нормалізація результату:**

Отримали результат:

Знак мантиси: 1 0 = 1.

Виконуємо зсув результату вліво, доки у першому розряді не опиниться одиниця, при цьому порядок числа зменшуємо на 1:

Mz  = 011110110000

11110110000 ← Pz=Pz-1=17-1=16=100002

Нормалізований результат:

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

**2.3 Третій спосіб множення.**

**2.3.1Теоретичне обгрунтування третього способу множення:**

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.

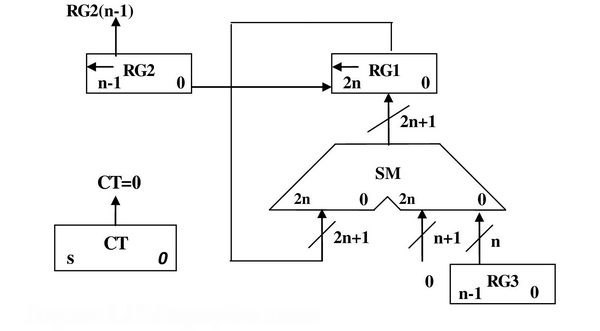
Множення мантис третім способом здійснюється зі старших розрядів множника, сума часткових добутків і множник зсуваються вліво, а множене нерухоме.

Z=Y+ Y…+ Y;

Z= Y+2(Y+2(Y…+2Y));

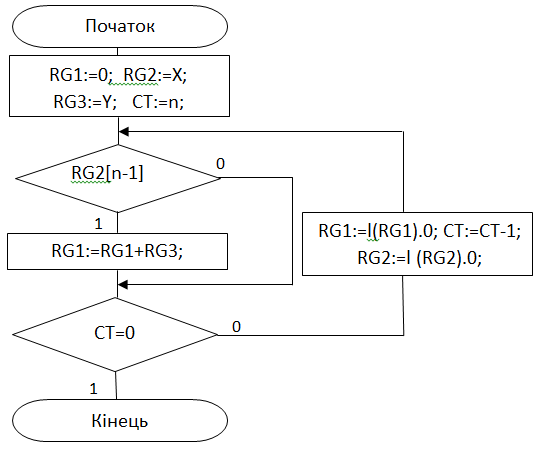
Z=;

**2.3.2 Операційна схема:**



*Рисунок 2.3.1 - Операційна схема*

**2.3.3 Змістовний мікроалгоритм:**



*Рисунок 2.3.2* **-** *Змістовний мікроалгоритм.*

**2.3.4 Таблиця станів регістрів**

Xпк = 1.111101 Yпк = 0.110000

*Таблиця 2.3.1- Таблиця станів регістрів*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | RG1 | RG2 | RG3 | CT |
| ПС | 000000000000 | 111101 | 000000110000 | 110 |
| 1 | 000000110000  000001100000 | 111010 | 000000110000 | 101 |
| 2 | +  000000110000  =  000101010000  001010100000 | 110100 | 000000110000 | 100 |
| 3 | +  000000110000  =  001011010000  010110100000 | 101000 | 000000110000 | 011 |
| 4 | +  000000110000  =  001011010000  010110100000 | 010000 | 000000110000 | 010 |
| 5 | 101101000000 | 100000 | 000000110000 | 001 |
| 6 | +  000000110000  =  **101101110000** | 000000 | 000000110000 | 000 |

**2.2.5 Обробка порядків:**

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків:

=1610; =610; =2210=101102

**2.2.6 Нормалізація результату:**

Отримали результат:

Знак мантиси: 1 0 = 1.

Зсуву робити не потрібно.

Mz  = 101101110000

Нормалізований результат:

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

**2.4 Четвертий спосіб множення.**

**2.4.1Теоретичне обґрунтування четвертого способу множення:**

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.

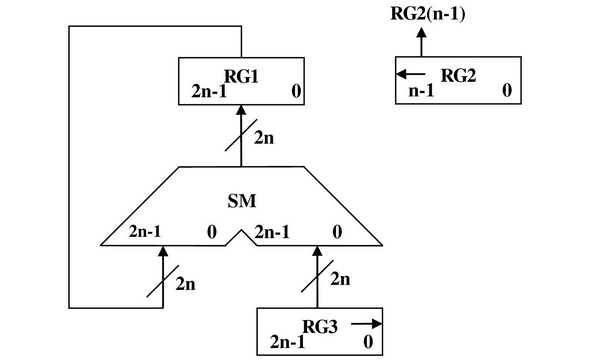
Множення здійснюється зі старших розрядів множника, сума часткових добутків залишається нерухомою, множене зсувається праворуч, множник ліворуч.

.

*.*

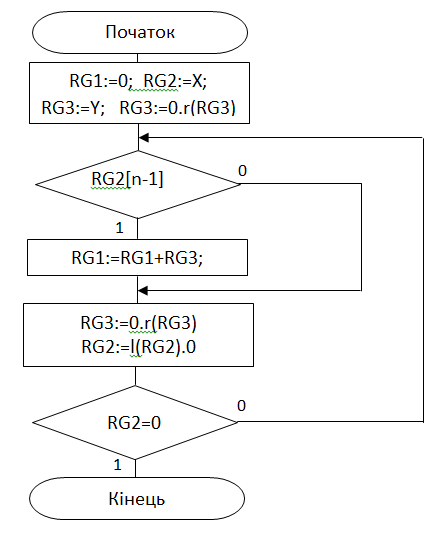
з початковими значеннями i=1, Y0=2-1Y, Z0=0.

**2.4.2 Операційна схема:**



*Рисунок 2.4.1-* *Операційна схема*

**2.4.3 Змістовний мікроалгоритм:**



*Рисунок 2.4.2* **-** *Змістовний мікроалгоритм*

**2.4.4 Таблиця станів регістрів:**

Xпк = 1. 101101 Yпк = 0.110000

*Таблиця 2.4.1- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № | RG1 | RG2 | RG3 |
| ПС | 000000000000 | 101101 | 011000000000 |
| 1 | 011000000000 | 011010 | 001100000000 |
| 2 | 011000000000 | 110100 | 000110000000 |
| 3 | +  000110000000  =  011110000000 | 101000 | 000011000000 |
| 4 | +  000011000000  =  100001000000 | 010000 | 000001100000 |
| 5 | 100001000000 | 100000 | 000000110000 |
| 6 | +  000000110000  =  **100001110000** | 000000 | 000000011000 |

**2.4.5 Обробка порядків:**

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків:

=2210; =610; =2810=111002

**2.4.6 Нормалізація результату:**

Отримали результат:

Знак мантиси: 1 0 = 1.

Зсуву робити не потрібно.

Mz  = 100001110000

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

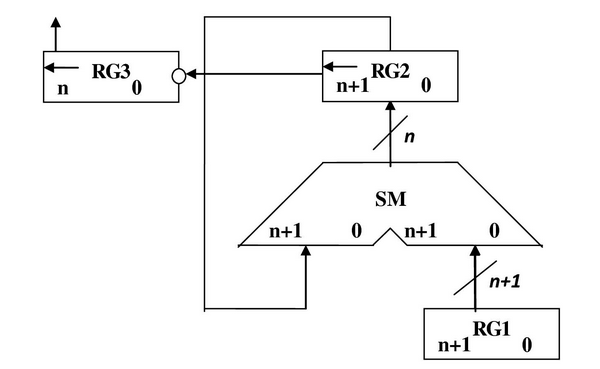
**2.5. Першиий спосіб ділення.**

**2.5.1Теоретичне 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.

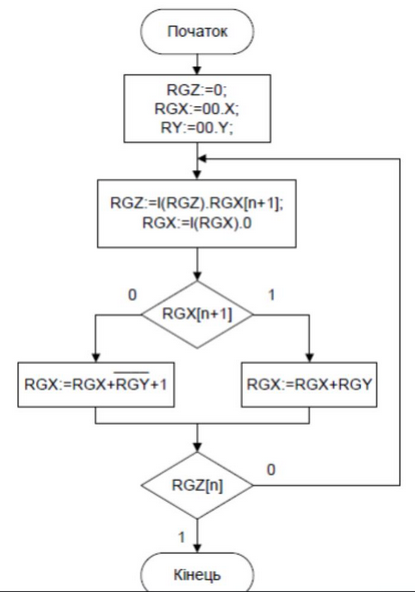
При реалізації ділення за першим методом здійснюється зсув вліво залишку при нерухомому дільнику. Черговий залишок формується в регістрі Р2 (у вихідному стані в цьому регістрі записаний Х). Виходи Р2 підключені до входів СМ безпосередньо, тобто ланцюги видачі коду з Р2 не потрібні. Час для підключення 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:**



*Рисунок 2.5.1-* *Операційна схема*

**2.5.3 Змістовний мікроалгоритм:**



*Рисунок 2.5.2-Змістовний мікроалгоритм*

**2.5.4 Таблиця станів регістрів:**

Xпк = 1.100001 Yпк = 0.110000

*Таблиця 2.5.1- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № | RG3(Z) | RG2(X) | RG1(Y) |
| ПС | 0000000 | 00100001 | 00110000 |
| 1 | 0000001 | 01000010  +  11010000  =  00010010 | 00110000 |
| 2 | 0000010 | 00100100  +  11010000  =  11110100 | 00110000 |
| 3 | 0000101 | 11101000  +  00110000  =  00011000 | 00110000 |
| 4 | 0001010 | 00110000  +  11010000  =  00000000 | 00110000 |
| 5 | 0010100 | 00000000  +  11010000  =  11010000 | 00110000 |
| 6 | 0101001 | 10100000  +  00110000  =  11010000 | 00110000 |
| 7 | **1010011** | 10100000  +  00110000  =  11010000 | 00110000 |

**2.5.5 Обробка порядків:**

Порядок частки дорівнює:

У даному випадку =2810; =610; =2210=101102;

**2.5.6 Нормалізація результату:**

Отримали результат:

Знак мантиси: 1 0 = 1.

Нормалізація мантиси не потрібна.

Mz=10100112

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

**2.6. Другий спосіб ділення.**

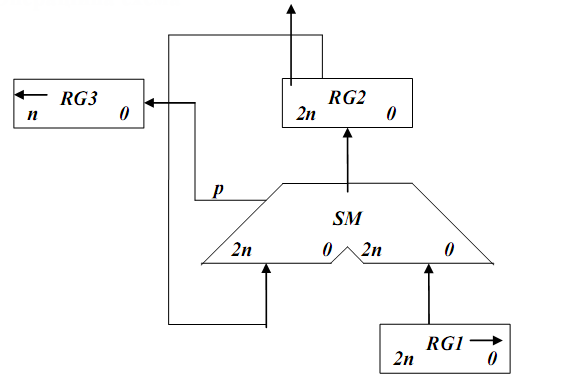
**2.6.1Теоретичне обгрунтування другого способу ділення:**

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**



*Рисунок 2.6.1-Операційна схема*

**2.6.3 Змістовний мікроалгоритм**

Початок

RG3:=0

RG1:=Y

RG2=X

RG2[2n+1]

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)

RG3[n]

1

0

1

0

1

*Рисунок 2.6.2-Змістовний мікроалгоритм*

**2.6.4 Таблиця станів регістрів**

Xпк = 1. 101001 Yпк = 0.110000

*Таблиця 2.6.1- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| № | RG3(Z) | RG2(X) | RG1(Y) |
| ПС | 0000001 | 010100100000 | 001100000000 |
| 1 | 0000011 | +  110100000000  =  001000100000 | 000110000000 |
| 2 | 0000111 | +  111010000000  =  000010100000 | 000011000000 |
| 3 | 0001110 | +  111101000000  =  111111100000 | 000001100000 |
| 4 | 0011101 | +  000001100000  =  000001000000 | 000000110000 |
| 5 | 0111011 | +  111111010000  =  000000010000 | 000000011000 |
| 6 | **1110110** | +  111111101000  =  111111111000 | 000000001100 |

**2.6.5 Обробка порядків:**

Порядок частки буде дорівнювати:

В моєму випадку =2210; =610; =162=100002;

**2.6.6 Нормалізація результату:**

Отримали результат:

Знак мантиси: 1 0 = 1.

Нормалізація мантиси не потрібна.

Mz=11101102

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

**2.7. Операція додавання чисел. Операція віднімання 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 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.

Для отримання різниці чисел знак мантиси від’ємника змінюється на протилежний.

1. Порівняння порядків.

Px=+1610=+100002

Py=+610=+01102

1610-610=1010=10102

P(заг)=10000

2. Вирівнювання порядків.

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

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, 110000 | 1010 | Початковий стан |
| 0, 011000 | 1001 | My= 0.r(My); ∆=∆-1 |
| 0, 001100 | 1000 | My= 0.r(My); ∆=∆-1 |
| 0, 000110 | 0111 | My= 0.r(My); ∆=∆-1 |
| 0, 000011 | 0110 | My= 0.r(My); ∆=∆-1 |
| 0, 000001 | 0101 | My= 0.r(My); ∆=∆-1 |
| 0, 000000 | 0100 | My= 0.r(My); ∆=∆-1 |
| …….. | …… | ………….. |
| 0, 000000 | 0000 | My= 0.r(My); ∆=∆-1 |

3. Додавання мантис у модифікованому ДК.

X мдк = 11.+1 =11. 000100+1 = 11. 000101

Yмдк = 00. 000000

*Таблиця 2.7.2-Додавання мантис у модифікованому ДК*

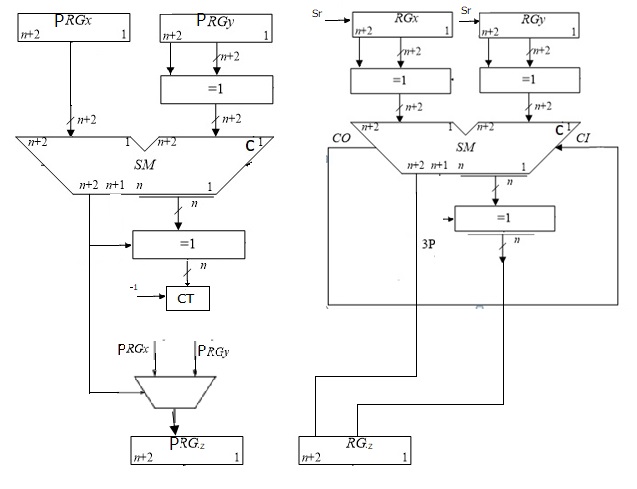
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MX | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| MY | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MZ | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |

Zмпк =1.111011

4. Нормалізація результату (в модифікованому прямому коді).

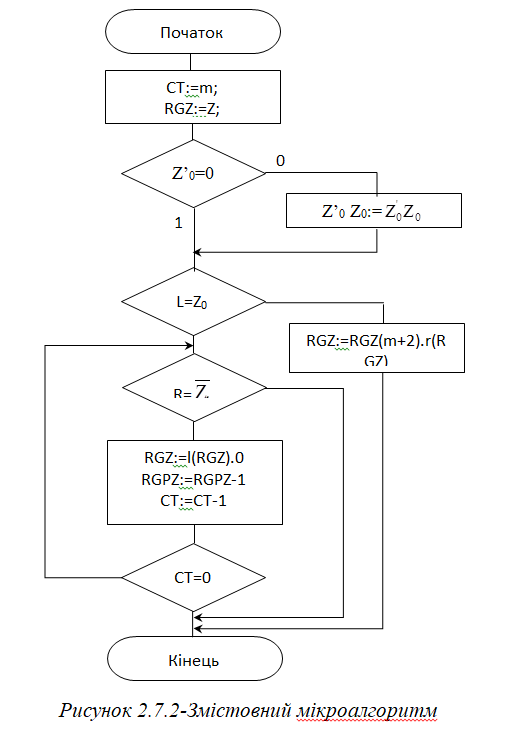
Для даного результату додавання нормалізація не потрібна.

**2.7.2 Операційна схема**

****

*Рисунок 2.7.1-Операційна схема* .

**2.7.3 Змістовний алгоритм** R



*Рисунок 2.7.2-Змістовний мікроалгоритм*

**2.7.4 Таблиця станів регістрів**

*Таблиця 2.7.5- Таблиця станів регістрів*

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **такту** | **RGPZ** | **RGZ** | ∆ |
| ПС | 0 | 0 |  |
| 1 | 0 | 0 |  |

**2.7.5 Форма запису результату з плаваючою комою**

Результат додавання Z=X+Y.

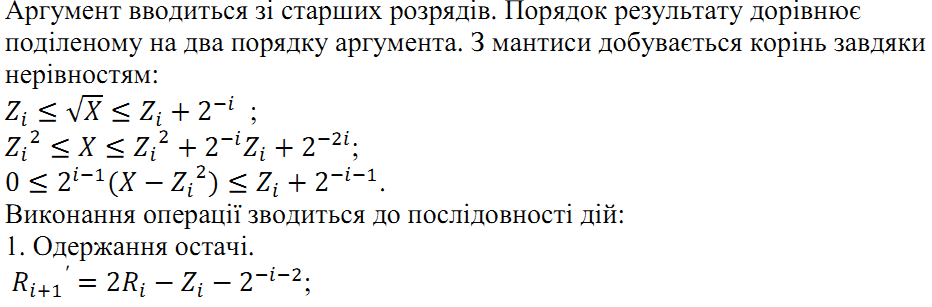
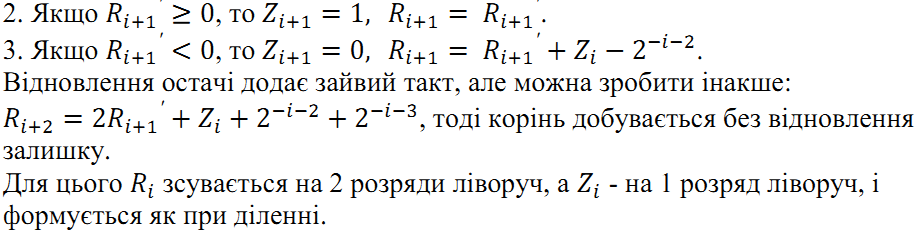
Zпк = 1. 111011

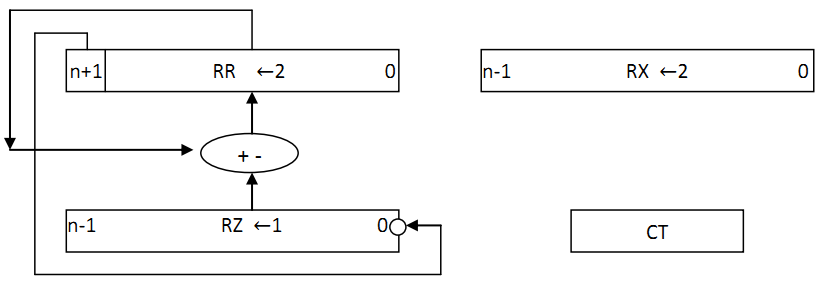
Pz = 1610 =100002 Mz = 1110112

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

**2.8.Операціядобування кореня**

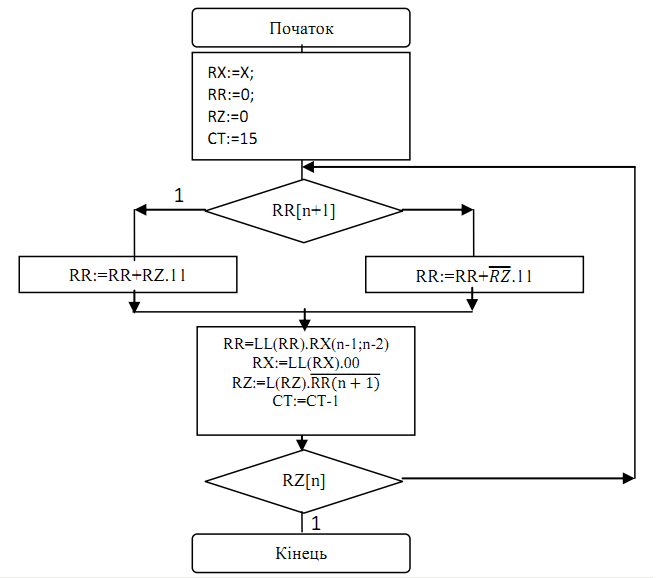
**2.8.1Теоритичнеобгрунтуванняоперації обчислення квадратного кореня**



**2.8.2 Операційна схема операції обчислення квадратного кореня**

*Рисунок 2.8.1 –Операційна схема*

**2.8.3 Змістовний мікроалгоритм**



6

*Рисунок 2.8.2 – Змістовний мікроалгоритм*

**2.8.4 Таблиця станів регістрів**

X = 10000 1110112

*Таблиця 2.8.1 – Таблиця станів регістрів*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | RZ | RR | RX | СТ |
| **пс** | 000000 | 00000000 | 111011 | 110 |
| **пз** | 00000011 |
| **1** | 000001 | 00000011  +  11111111  =  00000010  00001010 | 011101 | 101 |
| **2** | 000011 | 00001010  +  11111011  =  00000101  00000101  00010111 | 001110 | 100 |
| **3** | 000111 | 00010111  +  11110011  =  00001010  00101000 | 000111 | 011 |
| **4** | 001111 | 00101000  +  11100011  =  00001011  00101100 | 000011 | 010 |
| **5** | 011110 | 00101100  +  11000011  =  11101111  10111100 | 000001 | 001 |
| **6** | **111101** | 10111100  +  01111011  =  00110111  11011100 | 000000 | 000 |

**2.8.8 Обробка порядків**

В моєму випадку =8;

**2.8.9 Запис результату**

Отримали результат Z = **111101**

Результат нормалізований, готовий до запису у мантису:

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

**Завдання 3**

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;

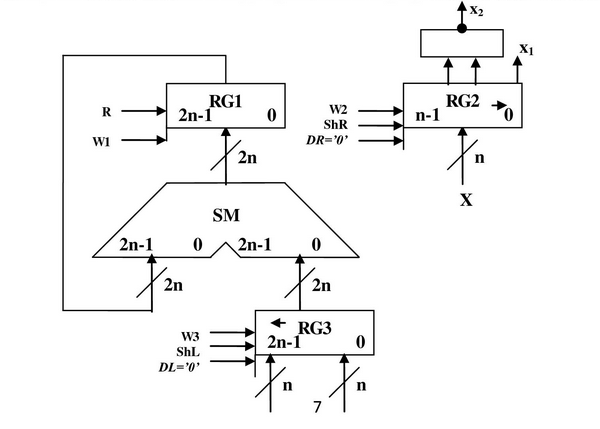
3.2 розробити закодований (структурний) мікроалгоритм (мікрооперації замінюються управляючими сигналами виду W,SL,SR тощо);

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.

x3x2x1 = 001 – для множення другим способом побудувати Мілі на RS-тригерах

**3.1Функціональна схема:**



*Рисунок 3.1- Функціональна схема.*

**3.2Закодований мікроалгоритм**

*Таблиця 2.2.2-Таблиця кодування операцій і логічних умов.*

|  |  |  |  |
| --- | --- | --- | --- |
| Кодування мікрооперацій | | 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 |

*Таблиця 3.1 Таблиця кодування сигналів пристрою множення другим способом*

|  |  |
| --- | --- |
| Входи операційного автомата | Виходи управляючого автомата |
| R,W2,W3, | Y1 |
| W1 | Y2 |
| ShR,ShL | Y3 |

Початок

Z1

R,W2,W3 (Y1)

Z2

0

X1

1

W1 (Y2)

Z3

ShR, ShL (Y3)

Z4

0

X2

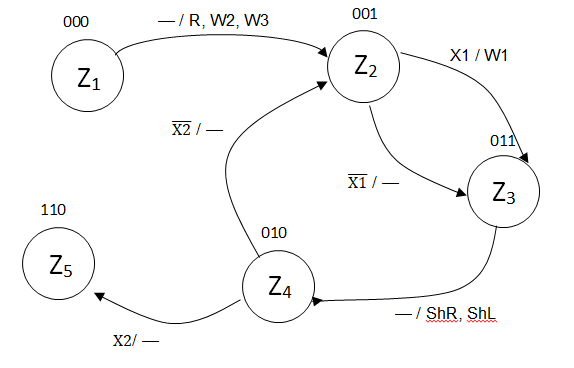
1

Z5

Кінець

*Рисунок 3.2-Закодований мікроалгоритм.*

**3.3 Граф управляючого автомата Мілі з кодами вершин:**



Y1

Y2

Y3

*Рисунок 3.3 - Граф автомата Мілі*

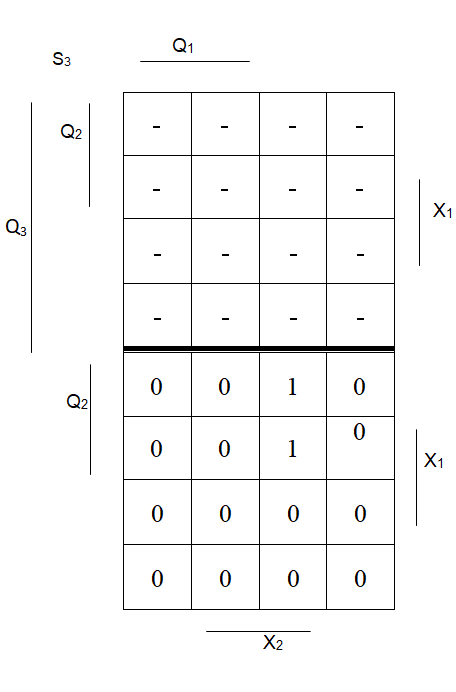
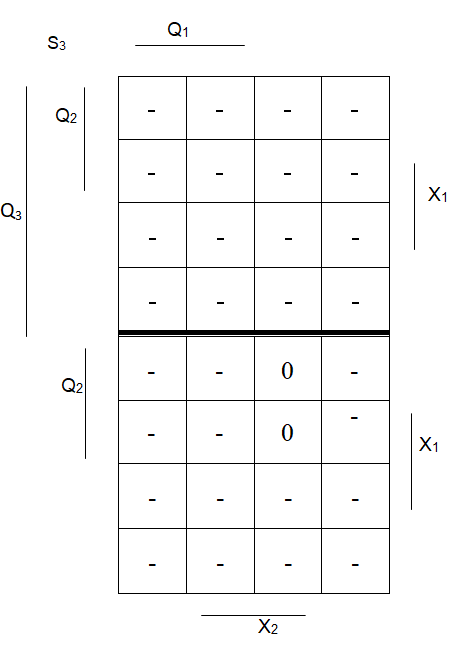
**3.4 Структурна таблиця автомата**

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

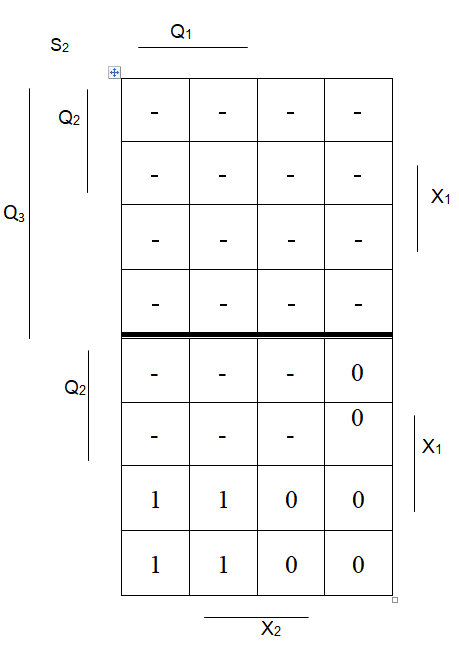
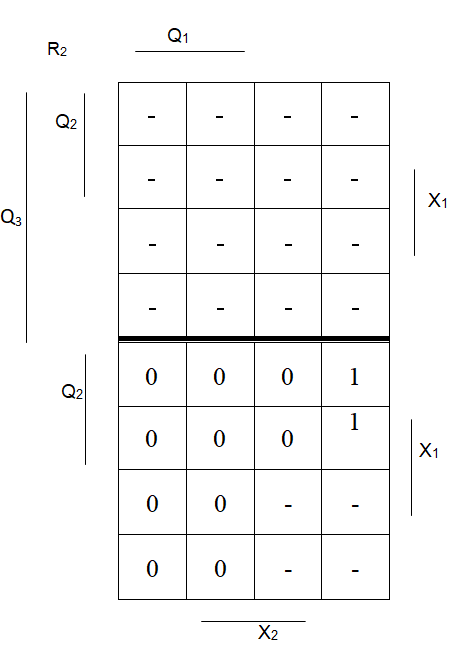
*Таблиця 3.2. Структурна таблиця автомата пристрою множення другим способом*

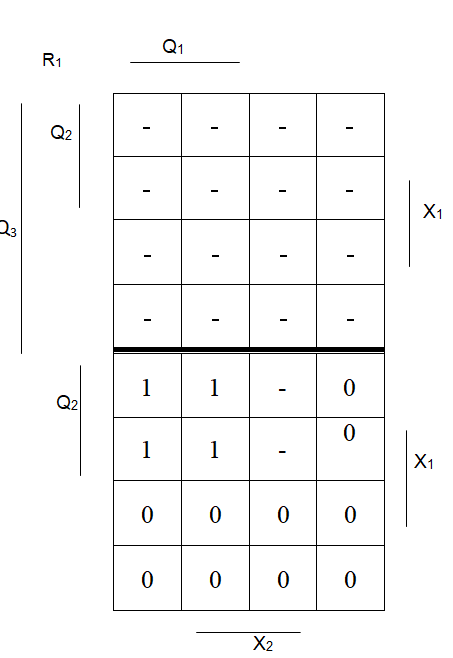
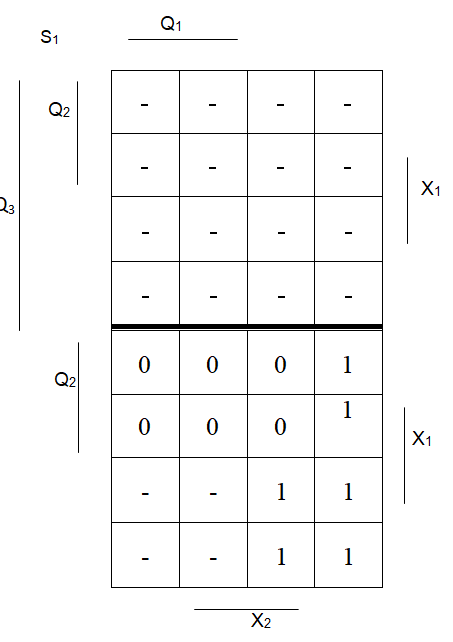
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Перехід | Q3Q2Q1 | Q3Q2Q1 | x1x2 | y1y2y3 | R3 | S3 | R2 | S2 | R1 | S1 |
| *z1z2* | 0 0 0 | 0 0 1 | - - | 1 0 0 | - | 0 | - | 0 | 0 | 1 |
| *z2 z3* | 0 0 1 | 0 1 1 | 1 - | 0 1 0 | - | 0 | 0 | 1 | 0 | - |
| *z2 z3* | 0 0 1 | 0 1 1 | 0 - | 0 0 0 | - | 0 | 0 | 1 | 0 | - |
| *z3 z4* | 0 1 1 | 0 1 0 | - - | 0 0 1 | - | 0 | 0 | - | 1 | 0 |
| *z4 z2* | 0 1 0 | 0 0 1 | - 0 | 0 0 0 | - | 0 | 1 | 0 | 0 | 1 |
| *z4 z5* | 0 1 0 | 1 1 0 | - 1 | 0 0 0 | 0 | 1 | 0 | - | - | 0 |

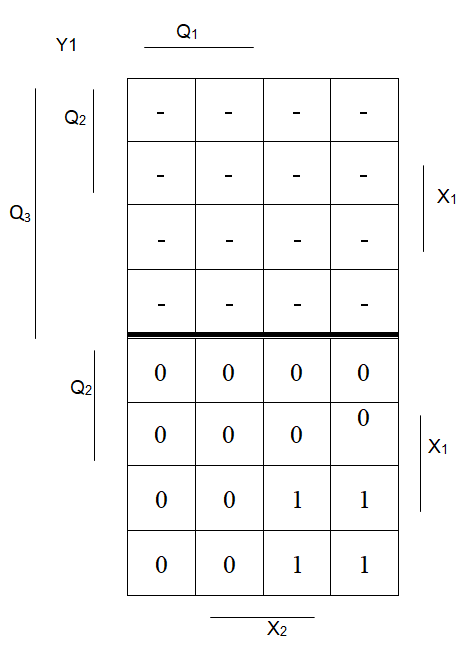
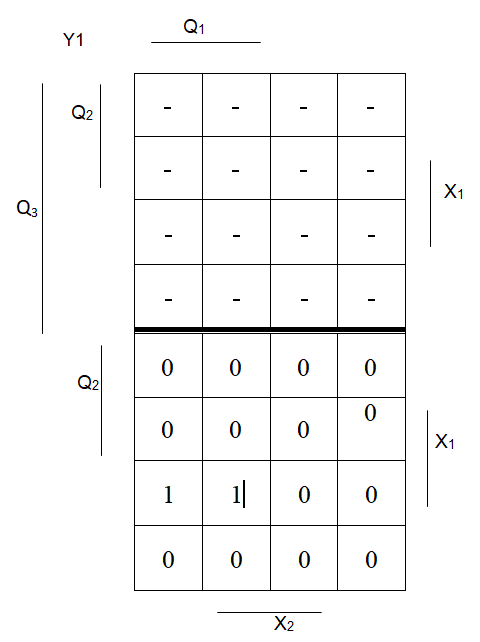
RS-тригер:



R



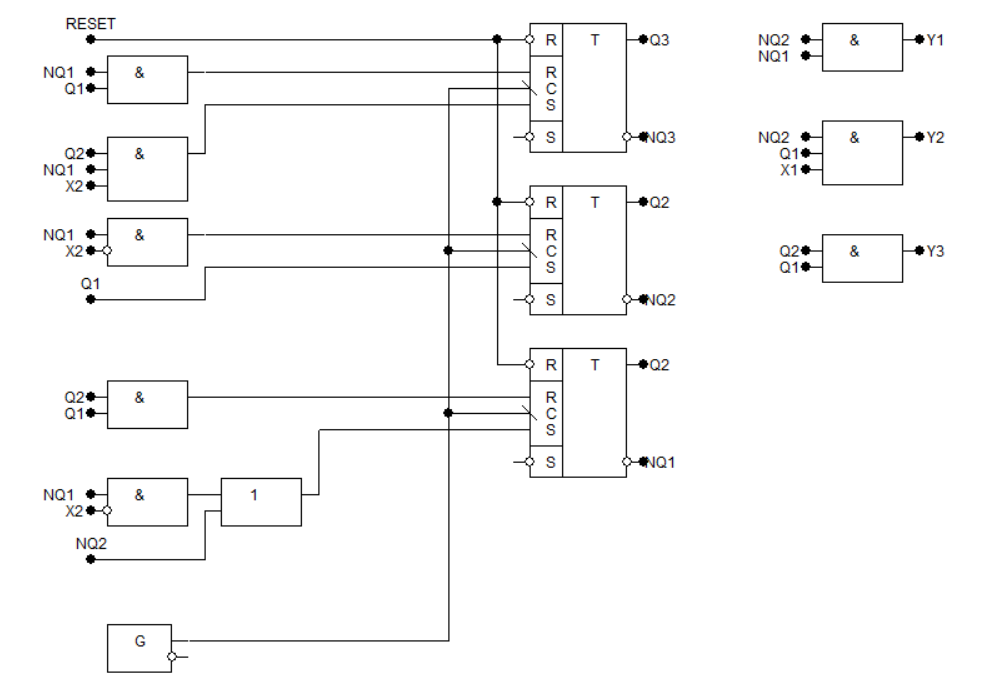


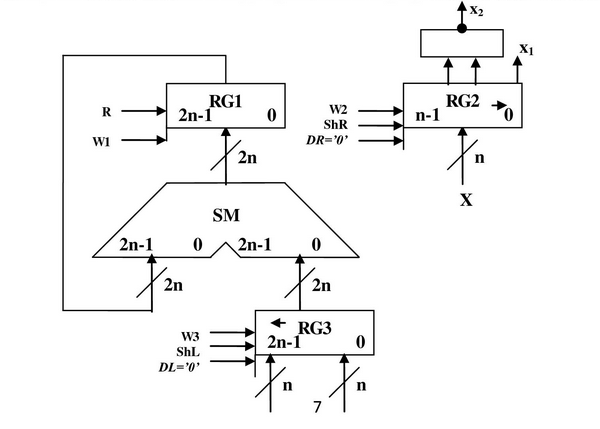


Y2

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

**підключені до входів операційного автомата)**





*Рисунок 3.4. Функціональна схема пристрою множення четвертим способом*

Y3

Y3

Y1

Y1

Y2

Y1

