НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

# "КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

#### ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

### Кафедра обчислювальної техніки

## РОЗРАХУНКОВА ГРАФІЧНА РОБОТА

з дисципліни ” Комп’ютерна логіка 2. Комп’ютерна арифметика ”

Виконав

Факультет ІОТ,

Група ІО-,

Залікова книжка № 4209

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

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

(підпис керівника)

Київ - 2015р.

**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 віднімання). Номери операцій (для п.3) відповідають порядку переліку (наприклад, 6 – ділення другим способом). Для обробки мантис кожної операції, подати:

2.1 теоретичне обґрунтування способу;

2.2 операційну схему;

2.3 змістовний мікроалгоритм;

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

2.5 функціональну схему з відображенням управляючих сигналів;

2.6 закодований мікроалгоритм (мікрооперації замінюються управл. сигналами);

2.7 граф управляючого автомата Мура з кодами вершин;

2.8 обробку порядків (показати у довільній формі);

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

Вказані пункти для операцій додавання та віднімання 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  побудувати управляючий автомат Мура на тригерах (тип вибрати самостійно) і елементах булевого базису.

**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).



**Завдання №1**



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

Px=; Mx= ;

Py=; My=;



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

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



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

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

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

E=Px+ ; m=7;

Ex=10000000 + 00001000=10001000;

E=Py+ ; m=7;

Ey=10000000 + 00000101=10000101;



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

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



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

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

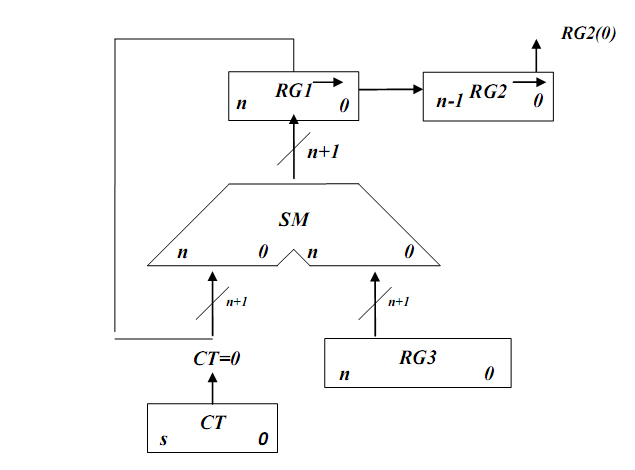
**Завдання №2**

**2.1 Перший спосіб множення.**

2.1.1 Теоретичне об**ґ**рунтування першого способу множення:

Під час множення першим способом в першому такті аналізується значення молодшого розряду регістру RG2(n), в якому знаходиться чергова цифра множника . Вміст регістра RG3 додається до вмісту регістра RG1 якщо значення молодшого розряду в другому регістрі рівне 1, або не додається якщо значення рівне 0. В другому такті здійснюється правий зсув в регістрах RG1 і RG2. За зсуву цифра молодшого розряду регістра РГ1 записується на місце звільненого старшого розряду RG2. Після виконання останнього такту молодші розряди добутку будуть записані у RG2, а старші в RG1.

2.1.2 Операційна схема:



*Рисунок 2.1.1 – Операційна схема*

2.1.2 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 виконання операції множення першим способом.*

2.1.4 Таблиця станів регістрів:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | RG1 | RG2 | RG3 | CT |
| П.С. | 0000000000000000 | 100101111000101 | 101001111000011 | 1111 |
| 1 | 0010100111100001 | 110010111100010 | 101001111000011 | 1110 |
| 2 | 0001010011110000 | 111001011110001 | 101001111000011 | 1101 |
| 3 | +  0101001111000011  =  0110100010110011  0011010001011001 | 111100101111000 | 101001111000011 | 1100 |
| 4 | 0001101000101100 | 111110010111100 | 101001111000011 | 1011 |
| 5 | 0000110100010110 | 011111001011110 | 101001111000011 | 1010 |
| 6 | 0000011010001011 | 001111100101111 | 101001111000011 | 1001 |
| 7 | +  0101001111000011  =  0101101001001110  0010110100100111 | 000111110010111 | 101001111000011 | 1000 |
| 8 | +  0101001111000011  =  1000000011101010  0100000001110101 | 000011111001011 | 101001111000011 | 0111 |
| 9 | +  0101001111000011  =  1001010000111000  0100101000011100 | 000001111100101 | 101001111000011 | 0110 |
| 10 | +  0101001111000011  =  1001110111011111  0100111011101111 | 100000111110010 | 101001111000011 | 0101 |
| 11 | 0010011101110111 | 110000011111001 | 101001111000011 | 0100 |
| 12 | +  0101001111000011  =  0111101100111010  0011110110011101 | 011000001111100 | 101001111000011 | 0011 |
| 13 | 0001111011001110 | 101100000111110 | 101001111000011 | 0010 |
| 14 | 0000111101100111 | 010110000011111 | 101001111000011 | 0001 |
| 15 | +  0101001111000011  =  **011000110010101**  **001011000001111** | 001011000001111 | 101001111000011 | 0000 |

2.1.5 Функціональна схема з відображенням управляючих сигналів:

***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

R, W2, W3, WCT

Z2

1

X1

Z3

W1

ShR1,ShR2,dec

Z4

X2

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, 011000110010101 <=; ;

Mz = 0, 110001100101010 ; .

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

**2.2 Другий спосіб множення**

2.2.1 Теоретичне об**ґ**рунтування другого способу множення:

Перед початком Х записують у RG2 , а У записують у молодші розряди регістру RG3. В кожному циклі множення додаванням кодів у третьому і першому регістрах керує цифра в RG2(n), а регістр RG3 зсувається на один вліво у результаті чого формується значення У. Оскільки сума часткових добутків у процесі множення нерухома , зсув у третьому регістрі можна сполучити з підсумовуванням. Завершується операція за визначенням нульового вмісту у регістрі RG2.

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*

2.1.4 Таблиця станів регістрів:

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | RG1 | RG2🡪 | RG3🡨 | CT |
| П.С. | 0 | 100101111000101 | 101001111000011 | 1111 |
| 1 | 000000000000000101001111000011 | 010010111100010 | 000000000000001010011110000110 | 1110 |
| 2 | 000000000000000101001111000011 | 001001011110001 | 000000000000010100111100001100 | 1101 |
| 3 | +  000000000000010100111100001100  =  000000000000011010001011001111 | 000100101111000 | 000000000000101001111000011000 | 1100 |
| 4 | 000000000000011010001011001111 | 000010010111100 | 000000000001010011110000110000 | 1011 |
| 5 | 000000000000011010001011001111 | 000001001011110 | 000000000010100111100001100000 | 1010 |
| 6 | 000000000000011010001011001111 | 000000100101111 | 000000000101001111000011000000 | 1001 |
| 7 | 000000000101001111000011000000  =  000000000101101001001110001111 | 000000010010111 | 000000001010011110000110000000 | 1000 |
| 8 | 000000001010011110000110000000  =  000000010000000111010100001111 | 000000001001011 | 000000010100111100001100000000 | 0111 |
| 9 | 000000010100111100001100000000  =  000000100101000011100000001111 | 000000000100101 | 000000101001111000011000000000 | 0110 |
| 10 | 000000101001111000011000000000  =  000001001110111011111000001111 | 000000000010010 | 000001010011110000110000000000 | 0101 |
| 11 | 000001001110111011111000001111 | 000000000001001 | 000010100111100001100000000000 | 0100 |
| 12 | +  000010100111100001100000000000  =  000011110110011101011000001111 | 000000000000100 | 000101001111000011000000000000 | 0011 |
| 13 | 000011110110011101011000001111 | 000000000000010 | 001010011110000110000000000000 | 0010 |
| 14 | 000011110110011101011000001111 | 000000000000001 | 010100111100001100000000000000 | 0001 |
| 15 | 010100111100001100000000000000  =  **011000110010101001011000001111** | 000000000000000 | 101001111000011000000000000000 | 0000 |

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

X2

X2

1

1

Z1

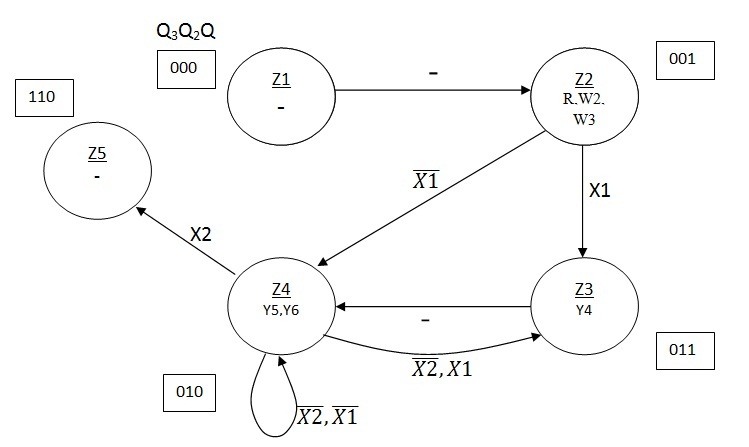
Z2

Z3

Z4

Z5

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, 011000110010101 <=; ;

Mz = 0, 110001100101010 ; .

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 |

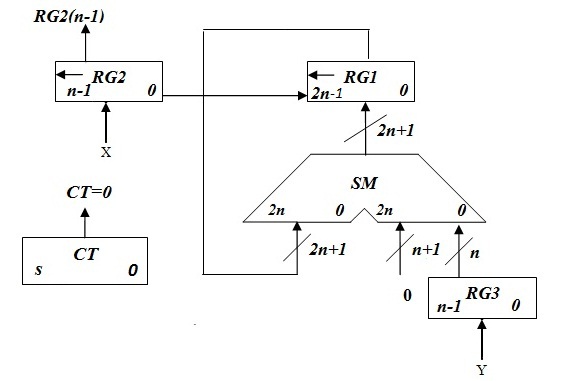
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |

**2.3 Третій спосіб множення.**

2.3.1Теоретичне обгрунтування третього способу множення:

На початку кожного циклу множення виконується зсув у першому та другому регістрах, а потім виконується додавання, яким керує 1 або нуль записані у старших розрядах другого регістра. У результаті підсумовування першого та третього регістрів може виникнути перенос у молодший розряд другого регістра. Після закінчення множення молодші розряди добутку будуть знаходитися в першому регістрі, а старші – другому.

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

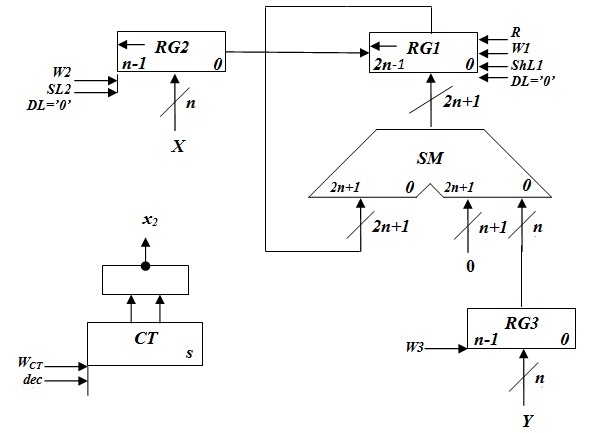
*Рисунок 2.3.2. Змістовний мікроалгоритм пристрою множення третім способом*

2.3.4 Таблиця станів регістрів

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1 | RG3 | RG2 | СТ |
| П.С. | 000000000000001010011110000110 | 101001111000011 | 100101111000101 | 1111 |
| 1 | 000000000000010100111100001100 | 101001111000011 | 001011110001010 | 1110 |
| 2 | 000000000000101001111000011000 | 101001111000011 | 010111100010100 | 1101 |
| 3 | 000000000000101001111000011000 | 101001111000011 | 101111000101000 | 1100 |
| 4 | +  000000000000000101001111000011  =  000000000000101111000111011011  000000000001011110001110110110 | 101001111000011 | 011110001010000 | 1011 |
| 5 | 000000000010111100011101101100 | 101001111000011 | 111100010100000 | 1010 |
| 6 | +  000000000000000101001111000011  =  000000000011000001101100101111  000000000110000011011001011110 | 101001111000011 | 111000101000000 | 1001 |
| 7 | +  000000000000000101001111000011  =  000000000110001000101000100001  000000001100010001010001000010 | 101001111000011 | 110001010000000 | 1000 |
| 8 | +  000000000000000101001111000011  =  000000001100010110100000000101  000000011000101101000000001010 | 101001111000011 | 100010100000000 | 0111 |
| 9 | +  000000000000000101001111000011  =  000000011000110010001111001101  000000110001100100011110011010 | 101001111000011 | 000101000000000 | 0110 |
| 10 | 000001100011001000111100110100 | 101001111000011 | 001010000000000 | 0101 |
| 11 | 000011000110010001111001101000 | 101001111000011 | 010100000000000 | 0100 |
| 12 | 000110001100100011110011010000 | 101001111000011 | 101000000000000 | 0011 |
| 13 | +  000000000000000101001111000011  =  000110001100101001000010010011  001100011001010010000100100110 | 101001111000011 | 010000000000000 | 0010 |
| 14 | 011000110010100100001001001100 | 101001111000011 | 100000000000000 | 0001 |
| 15 | +  000000000000000101001111000011  =  **011000110010101001011000001111** | 101001111000011 | 000000000000000 | 0000 |

2.3.5 Функціональна схема з відображенням управляючих сигналів

**

*Рисунок 2.3.3 Функціональна схема пристрою множення третім способом*

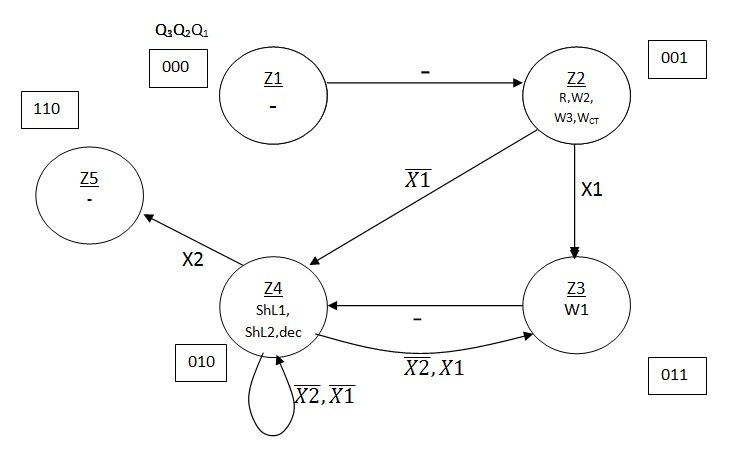
2.3.6 Закодований мікроалгоритм

*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, 011000110010101

Mz = 0, 110001100101010 ; .

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 |

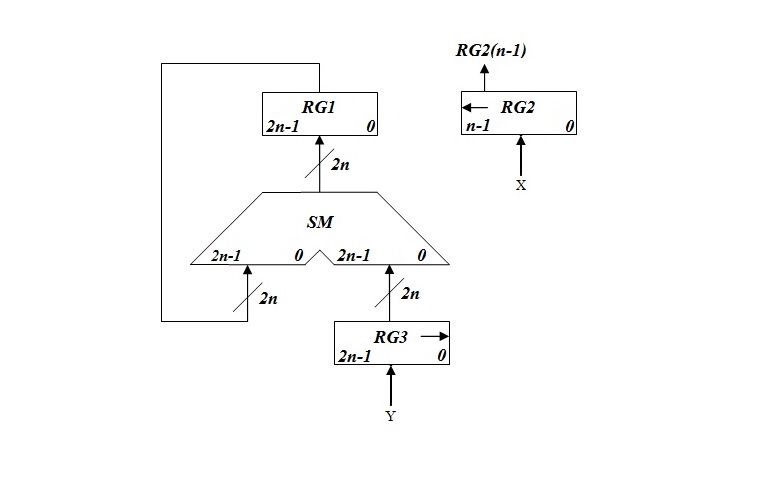
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |

**2.4 Четвертий спосіб множення.**

2.4.1 Теоретичне об**ґ**рунтування четвертого способу множення:

Перед початком множення множник записують у другий регістр, а множене – в старші розряди третього. Під час кожного такту цифра, що знаходиться у старшому розряді другого регістра керує, за встановленої там одиниці, підсумовуванням, а в третьому регістрі відбувається правий зсув числа на один розряд. Завершується операція за визначенням нульового вмісту у регістрі RG2.

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

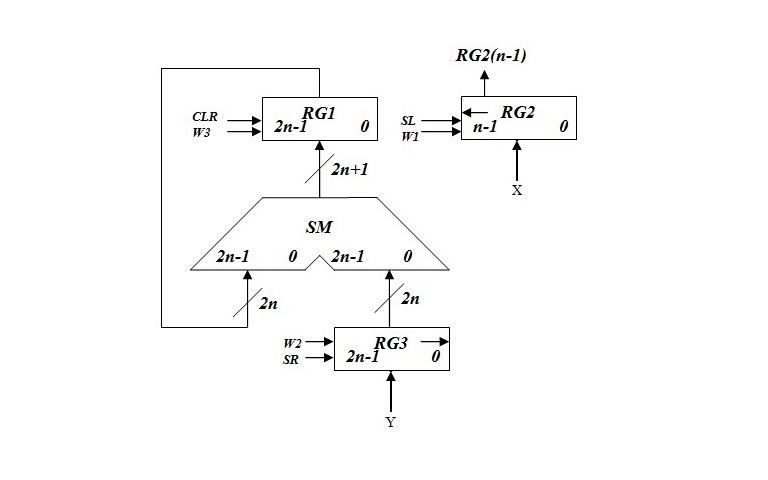
*Рисунок 2.4.2. Змістовний мікроалгоритм пристрою множення четвертим способом*

2.4.4 Таблиця станів регістрів

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

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 |
| П.С. | 000000000000000000000000000000 | 100101111000101 | 101001111000011 |
| 1 | 010100111100001100000000000000 | 001010011110000110000000000000 | 001011110001010 |
| 2 | 010100111100001100000000000000 | 000101001111000011000000000000 | 010111100010100 |
| 3 | 010100111100001100000000000000 | 000010100111100001100000000000 | 101111000101000 |
| 4 | +  000010100111100001100000000000  =  010111100011101101100000000000 | 000001010011110000110000000000 | 011110001010000 |
| 5 | 010111100011101101100000000000 | 000000101001111000011000000000 | 111100010100000 |
| 6 | +  000000101001111000011000000000  =  011000001101100101111000000000 | 000000010100111100001100000000 | 111000101000000 |
| 7 | +  000000010100111100001100000000  =  011000100010100010000100000000 | 000000001010011110000110000000 | 110001010000000 |
| 8 | +  000000001010011110000110000000  =  011000101101000000001010000000 | 000000000101001111000011000000 | 100010100000000 |
| 9 | +  000000000101001111000011000000  =  011000110010001111001101000000 | 000000000010100111100001100000 | 000101000000000 |
| 10 | 011000110010001111001101000000 | 000000000001010011110000110000 | 001010000000000 |
| 11 | 011000110010001111001101000000 | 000000000000101001111000011000 | 010100000000000 |
| 12 | 011000110010001111001101000000 | 000000000000010100111100001100 | 101000000000000 |
| 13 | +  000000000000010100111100001100  =  011000110010100100001001001100 | 000000000000001010011110000110 | 010000000000000 |
| 14 | 011000110010100100001001001100 | 000000000000000101001111000011 | 100000000000000 |
| 15 | +  000000000000000101001111000011  =  **011000110010101001011000001111** | 000000000000000010100111100001 | 000000000000000 |

2.4.5 Функціональна схема з відображенням управляючих сигналів

**

*Рисунок 2.4.3. Функціональна схема пристрою множення четвертим способом*

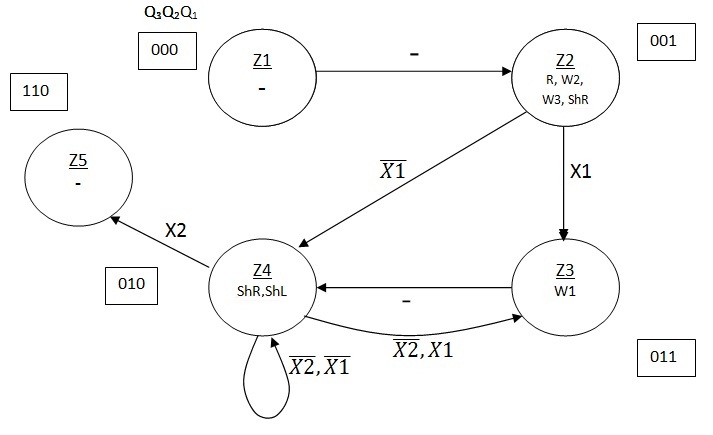
2.4.6 Закодований мікроалгоритм

*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, 011000110010101

Mz = 0, 110001100101010 ; .

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

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

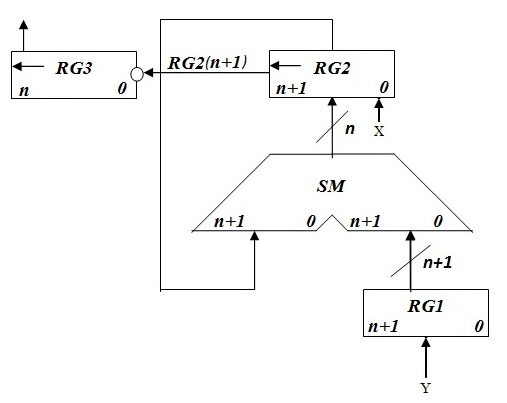
2.5.1 Теоретичне обґрунтування способу

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;

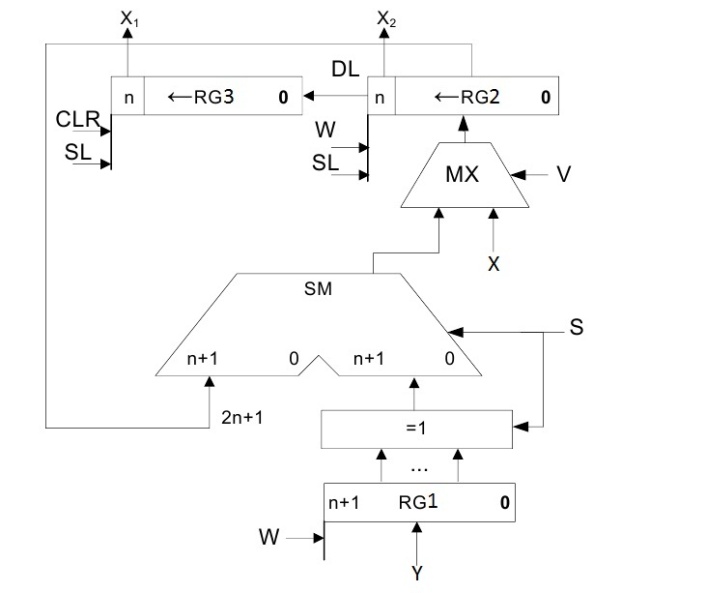
*Рисунок 2.5.2. Змістовний мікроалгоритм пристрою ділення першим способом*

2.5.4 Таблиця станів регістрів

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

|  |  |  |  |
| --- | --- | --- | --- |
| 6 | RG1 | RG2 | RG3 |
| П.С. | 0000000000000000 | 00100101111000101 |  |
| 1 | 0000000000000001 | 01001011110001010 +  11010110000111101 =  00100001111000111 |  |
| 2 | 0000000000000011 | 01000011110001110 +  11010110000111101 =  00011001111001011 |  |
| 3 | 0000000000000111 | 00110011110010110 +  11010110000111101 =  00001001111010011 |  |
| 4 | 0000000000001111 | 00010011110100110 +  11010110000111101 =  11101001111100011 |  |
| 5 | 0000000000011110 | 11010011111000110 +  00101001111000011 =  11111101110001001 |  |
| 6 | 0000000000111100 | 11111011100010010 +  00101001111000011 =  00100101011010101 |  |
| 7 | 0000000001111001 | 01001010110101010 +  11010110000111101 =  00100000111100111 |  |
| 8 | 0000000011110011 | 01000001111001110 +  11010110000111101 =  00011000000001011 |  |
| 9 | 0000000111100111 | 00110000000010110 +  11010110000111101 =  00000110001010011 |  |
| 10 | 0000001111001111 | 00001100010100110 +  11010110000111101 =  11100010011100011 |  |
| 11 | 0000011110011110 | 11000100111000110 +  00101001111000011 =  11101110110001001 |  |
| 12 | 0000111100111100 | 00001110110101010 +  11010110000111101 =  11100100111100111 |  |
| 13 | 0001111001111001 | 00001110110101010 +  11010110000111101 =  11100100111100111 |  |
| 14 | 0011110011110010 | 11001001111001110 +  00101001111000011 =  11110011110010001 |  |
| 15 | 0111100111100100 | 11100111100100010 +  00101001111000011 =  00010001011100101 |  |
| 16 | **1.111001111001001** | 00100010111001010 +  11010110000111101 =  11111001000000111 |  |

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

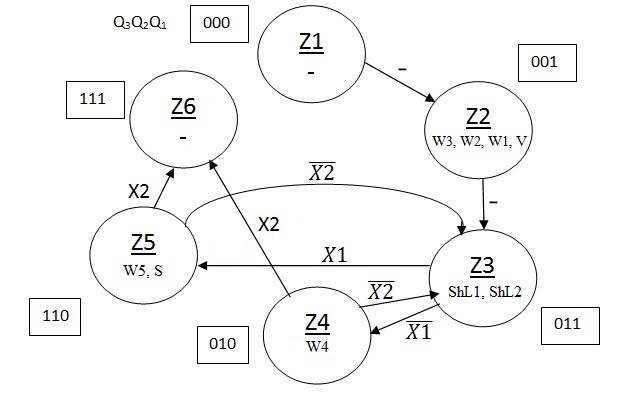
Z4

Z6

Z5

*Рисунок 2.5.4. Закодований мікроалгоритм пристрою ділення першим способом*

2.5.7Граф 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= 1.111001111001001

Знак мантиси: .

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

Зн.РZ PZ=+1210 Зн.МZ MZ

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

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

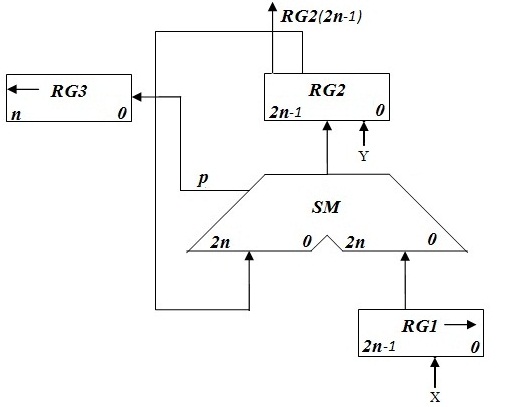
**2.6. Другий спосіб ділення.**

2.6.1 Теоритичне 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. Швидший за перший спосіб ділення.

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

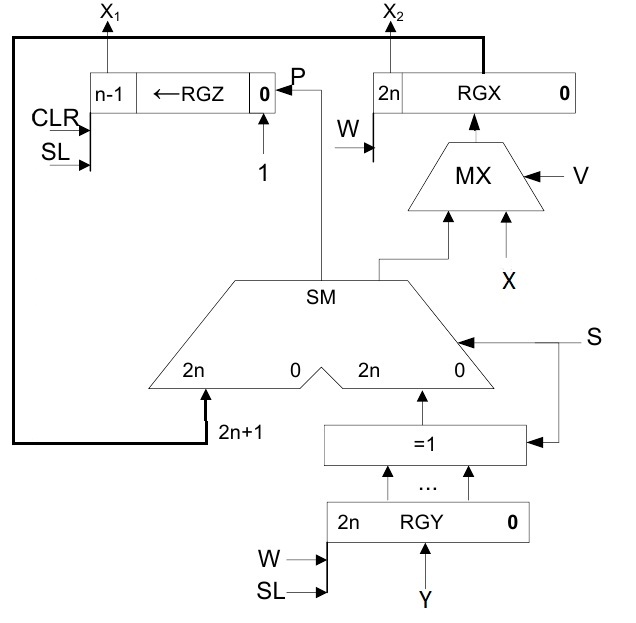
*Рисунок 2.6.2. Змістовний мікроалгоритм пристрою ділення другим способом*

2.6.4 Таблиця станів регістрів

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

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 |
| П.С. | 0000000000000001 | 010010111100010100000000000000 | 001010011110000110000000000000 |
| 1 | 0000000000000011 | 010010111100010100000000000000 +  110101100001111010000000000000 =  001000011110001110000000000000 | 000101001111000011000000000000 |
| 2 | 0000000000000111 | 001000011110001110000000000000 +  111010110000111101000000000000 =  000011001111001011000000000000 | 000010100111100001100000000000 |
| 3 | 0000000000001111 | 000011001111001011000000000000 +  111101011000011110100000000000 =  000000100111101001100000000000 | 000001010011110000110000000000 |
| 4 | 0000000000011110 | 000000100111101001100000000000 +  111110101100001111010000000000 =  111111010011111000110000000000 | 000000101001111000011000000000 |
| 5 | 0000000000111100 | 111111010011111000110000000000 +  000000101001111000011000000000 =  111111111101110001001000000000 | 000000010100111100001100000000 |
| 6 | 0000000001111001 | 111111111101110001001000000000 +  000000010100111100001100000000 =  000000010010101101010100000000 | 000000001010011110000110000000 |
| 7 | 0000000011110011 | 000000010010101101010100000000 +  111111110101100001111010000000 =  000000001000001111001110000000 | 000000000101001111000011000000 |
| 8 | 0000000111100111 | 000000001000001111001110000000 +  111111111010110000111101000000 =  000000000011000000001011000000 | 000000000010100111100001100000 |
| 9 | 0000001111001111 | 000000000011000000001011000000 +  111111111101011000011110100000 =  000000000000011000101001100000 | 000000000001010011110000110000 |
| 10 | 0000011110011110 | 000000000000011000101001100000 +  111111111110101100001111010000 =  111111111111000100111000110000 | 000000000000101001111000011000 |
| 11 | 0000111100111100 | 111111111111000100111000110000 +  000000000000101001111000011000 =  111111111111101110110001001000 | 000000000000010100111100001100 |
| 12 | 0001111001111001 | 111111111111101110110001001000 +  000000000000010100111100001100 =  000000000000000011101101010100 | 000000000000001010011110000110 |
| 13 | 0011110011110010 | 000000000000000011101101010100 +  111111111111110101100001111010 =  111111111111111001001111001110 | 000000000000000101001111000011 |
| 14 | 0111100111100100 | 111111111111111001001111001110 +  000000000000000101001111000011 =  111111111111111110011110010001 | 000000000000000010100111100001 |
| 15 | **1.111001111001001** | 111111111111111110011110010001 +  000000000000000010100111100001 =  000000000000000001000101110010 | 000000000000000001010011110000 |

2.6.5 Функціональна схема з відображенням управляючих сигналів



*Рисунок 2.6.3. Функціональна схема пристрою ділення другим способом*

2.6.6Закодований 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

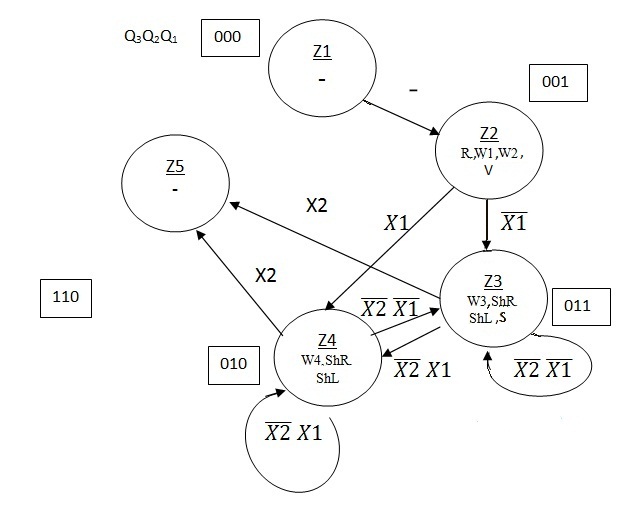
Z3

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= 1.111001111001001

Знак мантиси: .

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

Зн.РZ PZ=+1210 Зн.МZ MZ

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

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

**2.7 Операція додавання чисел**

2.7.1 Теоретичне об**ґ**рунтування 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.

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, 101001111000011 | 11 | П.С. |
| 0, 010100111100001 | 10 | 🡪 |
| 0, 001010011110000 | 01 | 🡪 |
| 0, 000101001111000 | 00 | 🡪 |

Додавання та віднімання мантис у модифікованому ДК

****

*Таблиця 2.7.2. Додавання мантис*

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

*.*



*Таблиця 2.7.2. Віднімання мантис*

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

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

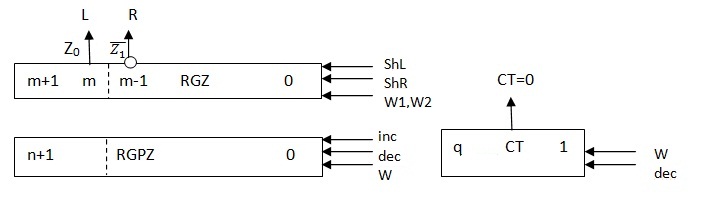
*Рисунок 2.7.2. Змістовний мікроалгоритм пристрою додавання двох чисел*

2.7.4Таблиця 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

**

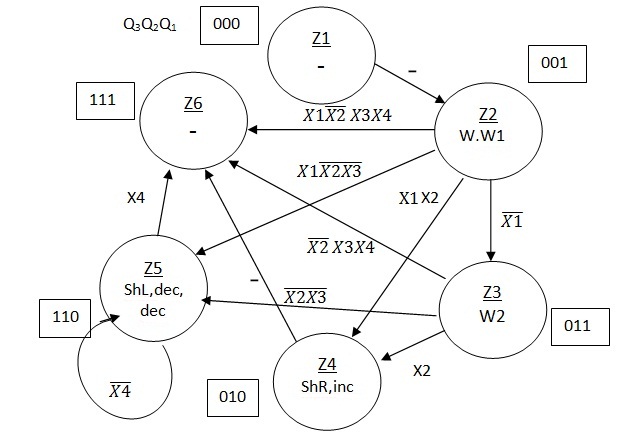
*Рисунок 2.7.4. Функціональна схема пристрою додавання двох чисел*

2.7.6Закодований 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

При додаванні мантис :

Зн.РZ PZ=+710 Зн.МZ  MZ

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

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

При відніманні мантис :

Зн.РZ PZ=+710 Зн.МZ  MZ

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

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

**3. управляючий автомат Мура на тригерах**

x3x2x1 +1=001=1 – операція множення першим способом.

RS-тригери.

3.1 Таблиця співвідношення управляючих входів операційного автомата і виходів управляючого автомата

За закодованим мікроалгоритмом складемо таблицю:

*Таблиця 3.1.1 Таблиця кодування сигналів*

|  |  |
| --- | --- |
| Входи операційного автомата | Виходи управляючого автомата |
| R, W2, W3, WCT | Y1 |
| W1 | Y2 |
| SR1,SR2,D | Y3 |

3.2 Мікроалгоритм в термінах управляючого автомата

Зробимо автомат Мура циклічним задля зменшення кількості вершин.

Початок

Z1

Y1

Z2

X1

1

Z3

Y2

Z4

Y3

X2

1

Z1

Кінець

*Рисунок 5.1- Закодований мікроалгоритм*

Будуємо граф автомата Мура

00

01

Q1Q2

-

Z1

-

Z2

Y1

X1

X2

-

Z4

Y3

Z3

Y2

10

11

*Рисунок 5.2- Граф автомата Мура*

3.3 Структурна таблиця автомата

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

*Таблиця 3.2.1-Структурна таблиця автомата*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Перехід | Q2Q1 | Q2Q1 | x1x2 | y1y2y3 | R2 | S2 | R1 | S1 |
| *z1z2* | 00 | 01 | -- | 000 | - | 0 | 0 | 1 |
| *z2 z3* | 01 | 11 | 1- | 100 | 0 | 1 | 0 | - |
| *z2 z4* | 01 | 10 | 0- | 100 | 0 | 1 | 1 | 0 |
| *z3 z4* | 11 | 10 | -- | 010 | 0 | - | 1 | 0 |
| *z4 z3* | 10 | 11 | 10 | 001 | 0 | - | 0 | 1 |
| *z4 z1* | 10 | 00 | -1 | 001 | 1 | 0 | - | 0 |
| *z4 z4* | 10 | 10 | 00 | 001 | 0 | - | - | 0 |

RS:

3.4 Синтез функцій виходів і переходів

0

1

0

0

Y3

1

0

0

0

Y2

0

0

1

0

Q2

Q1

Y1

0

0

0

0

0

0

0

1

-

-

1

1

0

0

1

1

Q2

Q1

X2

X1

S1

1-

1

-

-

1

1-

-

0

0

0

0

0

1

1

0

0

Q2

Q1

X2

X1

R1

-

-

0

-

-

-

-

0

-

1

1

0

0

1

1

0

0

Q2

Q1

X2

X1

S2

0

0

1

0

0

0

1

0

0

0

-

-

0

0

-

-

Q2

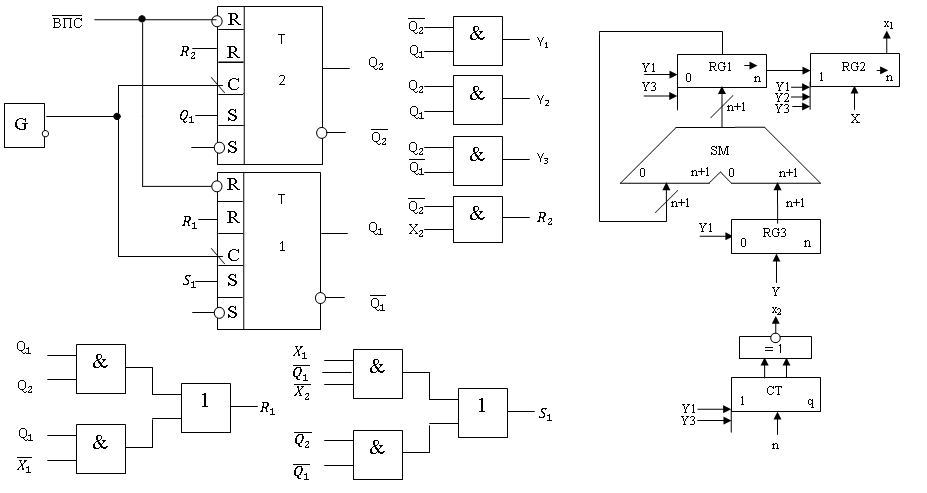
Q1

X2

X1

R2

*Рисунок 5.3- Діаграми Вейча*

****

*Рисунок 5.5- Функціональна схема пристрою*

**Висновок:**

Таким чином, виконуючи дану розрахункову роботу, я повторив такі операції над числами в двійковому коді , як множення, ділення та додавання. Для кожної операції була подана операційна схема, змістовний мікроалгоритм, таблиця станів регістрів, функціональна схема з відображенням управляючих сигналів, граф відповідного закодованого мікроалгоритму і ,власне, закодований мікроалгоритм. Був синтезований управляючий автомат для операційного пристрою множення першим способом, згідно з варіантом, відповідно, даний операційний пристрій був побудований на RS-тригерах.