Міністерство науки та освіти Україні

Національний технічний університет України «Київський політехнічний інститут»

**Розрахункова робота по курсу „КЛ-2”**

### Виконав

### студент І-го курсу ФІОТ гр. ІО-92

### Пинтя М.І.

### Керівник: Жабін В.І.

Київ 2010

**ЗМІСТ**

Завдання та обґрунтування варіанту……………………………………………5

1 Операція множення чисел……………………………………………………..6

1.1 Перший спосіб множення…………………………………………….7

1.1.1 Теоретичне обґрунтування способу…………………………7

1.1.2 Операційна схема……………………………………………..7

1.1.3 Змістовний мікроалгоритм…………………………………...7

1.1.4 Таблиця станів регістрів……………………………………...8

1.1.5 Функціональна схема пристрою……………………………..9

1.1.6 Закодований мікроалгоритм…………………………………9

1.1.7 Граф управляючого автомата……………………………….10

1.1.8 Обробка порядків ……………………………………………10

1.1.9 Форма подання результату в пам’ять………………………10

1.2 Другий спосіб множення……………………………………………..11

1.2.1 Теоретичне обґрунтування способу………………………...11

1.2.2 Операційна схема…………………………………………….11

1.2.3 Змістовний мікроалгоритм…………………………………..11

1.2.4 Таблиця станів регістрів……………………………………..12

1.2.5 Функціональна схема пристрою…………………………….12

1.2.6 Закодований мікроалгоритм…………………………………13

1.2.7 Граф управляючого автомата………………………………..13

1.2.8 Обробка порядків …………………………………………….14

1.2.9 Форма подання результату в пам’ятті……………………….15

1.3 Третій спосіб множення…………………………………………………….16

1.3.1 Теоретичне обґрунтування способу………………………...16

1.3.2 Операційна схема…………………………………………….16

1.3.3 Змістовний мікроалгоритм…………………………………..16

1.3.4 Таблиця станів регістрів……………………………………..17

1.3.5 Функціональна схема пристрою…………………………….18

1.3.6 Закодований мікроалгоритм……………………………...…19

1.3.7 Граф управляючого автомата…………………………...…..19

1.3.8 Обробка порядків ……………………………………...…….20

1.3.9 Форма подання результату в пам’ятті……………..……….20

1.4 Четвертий спосіб множення………………...…………………………..…..21

1.4.1 Теоретичне обґрунтування способу………………………...21

1.4.2 Операційна схема…………………………………………….21

1.4.3 Змістовний мікроалгоритм…………………………………..21

1.4.4 Таблиця станів регістрів……………………………………..22

1.4.5 Функціональна схема пристрою…………………………….22

1.4.6 Закодований мікроалгоритм…………………………………23

1.4.7 Граф управляючого автомата………………………………..23

1.4.8 Обробка порядків …………………………………………….24

1.4.9 Форма подання результату в пам’ятті……………………….24

2 Операція ділення чисел………………………………………………………...25

2.1 Перший спосіб…………………………………………………………25

2.1.1 Теоретичне обґрунтування способу…………………………25

2.1.2 Операційна схема……………………………………………..25

2.1.3 Змістовний мікроалгоритм…………………………………...26

2.1.4 Таблиця станів регістрів………………………………………26

2.1.5 Функціональна схема пристрою……………………………...27

2.1.6 Закодований мікроалгоритм…………………………………..28

2.1.7 Граф управляючого автомата…………………………………28

2.1.8 Обробка порядків ……………………………………………..28

2.1.9 Форма подання результату в пам’ятті………………………..28

2.2 Другий спосіб…………………………………………………………...29

2.2.1 Теоретичне обґрунтування способу…………………………..29

2.2.2 Операційна схема………………………………………………29

2.2.3 Змістовний мікроалгоритм…………………………………….30

2.2.4 Таблиця станів регістрів……………………………………….30

2.2.5 Функціональна схема пристрою………………………………31

2.2.6 Закодований мікроалгоритм…………………………………...32

2.2.7 Граф управляючого автомата………………………………….32

2.2.8 Обробка порядків ……………………………………………....33

2.2.9 Форма подання результату в пам’ятті………………………...33

3 Операція додавання чисел

3.1 Теоретичне обґрунтування способу……………………………..34

3.2 Операційна схема…………………………………………………34

3.3 Змістовний мікроалгоритм……………………………………….35

3.4 Таблиця станів регістрів………………………………………….35

3.5 Функціональна схема пристрою…………………………………36

3.6 Закодований мікроалгоритм……………………………………...36

3.7 Граф управляючого автомата…………………………………….37

3.8 Обробка порядків ………………………………………………....37

3.9 Форма подання результату в пам’ятті……………………...…...37

4 Операція додавання чисел

4.1 Теоретичне обґрунтування способу……………………………..38

4.2 Операційна схема…………………………………………………38

4.3 Змістовний мікроалгоритм……………………………………….39

4.4 Таблиця станів регістрів………………………………………….39

4.5 Функціональна схема пристрою…………………………………40

4.6 Закодований мікроалгоритм……………………………………...41

4.7 Граф управляючого автомата…………………………………….41

4.8 Обробка порядків …………………………………………………41

4.9 Форма подання результату в пам’ятті…………………………...41

5 Синтез управляючого автомата для операційного пристрою (дати назву пристрою)

5.1 Таблиця співвідношення управляючих входів операційного автомата і виходів управляючого автомата

……………………………………………………..42

5.2 Мікроалгоритм в термінах управляючого автомата…………………...42

5.3 Структурна таблиця автомата…………………………………………...43

5.4 Синтех функцій виходів і переходів……………………………………43

5.5 Функціональна схема пристою (виходи управляючого автомата підключені до входів операційного автомата)……………………………………………...45

1) Числа

**Завдання:**

*X* і *Y* в прямому коді записати у формі з плаваючою комою (з порядком і

2

2

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). (0,5)

2) Виконати 8 операцій з числами

2

2

*X* і *Y* з плаваючою комою (чотири способи

множення, два способи ділення, додавання та добування кореня з

2

*X* ). Номери

операцій (для п.3) відповідають порядку переліку (наприклад, 6 – ділення другим способом).

Для обробки мантис кожної операції, подати:

2.1) теоретичне обґрунтування способу; (0,2)

2.2) операційну схему; (0,2)

2.3) змістовний мікроалгоритм; (0,2)

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

основних розрядів мантиси результату; (1,5)

2.5) функціональну схему з відображенням управляючих сигналів; (0.5)

2.6) закодований мікроалгоритм (мікрооперації замінюються управл. сигналами);

(0.3)

2.7) граф управляючого автомата Мура з кодами вершин; (0,5)

2.8) обробку порядків (показати у довільній формі); (0,5)

2.9) форму запису нормалізованого результату з плаваючою комою в пам’ять. (0,1)

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

урахуванням можливого нулевого результату.

3) Для операції з номером

*x*3 *x*2 *x*1

побудувати управляючий автомат Мура на тригерах

(тип вибрати самостійно) і елементах булевого базису. (1,5)

**Обґрунтування варіанту**

Номер залікової книжки: 922010 = 100100000001002

X2 = –10010001,0001100

Y2 = +10100,0010001001

**Виконання роботи**

***Завдання 1***

В прямому коді:

**X2:**

Зн.Р P=+810 Зн.М M= -,100100010001100

### Y2:

Зн.Р P=+510 Зн.М M= +,101000010001001

1 **Операції множення чисел**

1.1 Перший спосіб множення

1.1.1 Вираз Z = X Y = Yx12-1 + Yx22-2 +…+ Yxi2-i +…+ Yxn2-n можна подати у вигляді:

Zi = (Zi-1 + Yxn-i+1)2-1 з початковими значеннями i = 1, Z0 = 0, причому Zn = Z = Y\*X.

У розглянутому способі множення здійснюється з молодших розрядів множника, сума

часткових добутків зсувається вправо, а множене залишається нерухомим.

1.1.2 Операційна схема:

DR=0

DR

n RG1**→** 0 n-1 RG2**→** 0

**+**

n-1 RG3**→** 0

S CT 0

Рис. 1.1 – Операційна схема множення І способом

1.1.3 Змістовний мікроалгоритм:

Початок

RG1:=0;

RG2:=X; RG3:=Y; CN:=n;

0

RG2=0

1

RG1:=RG1+ RG3

RG1:=0.r(RG1)

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

0

CT=0

1

Кінець

Рис. 1.2 – Мікроалгоритм множення І способом

1.1.4 Таблиця станів регістрів:

Табл. 2.1 – Таблиця станів регістрів для множення І способом

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1 | RG2 | RG3 | СТ |
| П.С. | 0000000000000000 | 100100010001100 | 010010001000110 | 1111 |
| 1🡪 | 0000000000000000 | 010010001000110 |  | 1110 |
| 2🡪 | 0010100001000100 | 001001000100011 |  | 1101 |
| 3  🡪 | +0101000010001001  0111100011001101  0011110001100110 | 100100100010001 |  | 1100 |
| 4🡪 | 0001111000110011 | 110010010001000 |  | 1011 |
| 5🡪 | 0000111100011001 | 011001001000100 |  | 1010 |
| 6🡪 | 0000011110001100 | 101100100100010 |  | 1001 |
| 7  🡪 | +0101000010001001  0101100000010101  0010110000001010 | 110110010010001 |  | 1000 |
| 8🡪 | 0001011000000101 | 111011001001000 |  | 0111 |
| 9🡪 | 0000101100000010 | 011101100100100 |  | 0110 |
| 10🡪 | 0000010110000001 | 101110110010010 |  | 0101 |
| 11🡪 | +0101000010001001  0101011000001010  0010101100000101 | 010111011001001 |  | 0100 |
| 12🡪 | 0001010110000010 | 001011101100100 |  | 0011 |
| 13🡪 | 0000101011000001 | 100101110110010 |  | 0010 |
| 14🡪 | 0000101100010010 | 010010111011001 |  | 0001 |
| 15  🡪 | +0101000010001001  0101101101001010  **0,010110110100101** | **001001011101100** |  | 0000 |

1.1.5 Функціональна схема:

X1

|  |  |  |
| --- | --- | --- |
|  | |  |
| n-1 RG2→ | 0 | |
|  | | |

DR=0

CLR W

DR

n RG1**→** 0

W

SR SR

**X**

SM

n 0 n 0 X2

n-1 RG3 0 1

W

**Y** W

dec

q CT 0

Рис. 1.3 – Функціональна схема з відображенням управляючих сигналів

1.1.6 Закодований мікроалгоритм:

Початок **Z1**

CLR1; W2;

W3; WCT; **Z2**

0

X1

1

W3 **Z3**

SR1; SR2; dec; **Z4**

0

X2

1

Кінець **Z5**

Рис. 1.4 – Закодований мікроалгоритм множення І способом

1.1.7 Граф управляючого автомата Мура:

**000**

**Z1 -**

**-**

**CLR1; W3; W2; WCT**

**001**

**Z5**

**- X2 X1 X1**

**111**

**Z4 -**

**SR1; SR2;**

**DEC;**

**Z3 W1**

**010**

**X1, X2**

**011**

**X1, X2**

Рис. 1.5 – Граф автомата на І спосіб множення

1.1.8 Обробка порядку результату:

.

Нормалізація мантиси.

MZ= , 0101101101001010 🡨 , .

, 101101101001010 .

Знак мантиси: .

1.1.9 Запис результату в пам'ять

**Зн.Р P=+310 Зн.М M**

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

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

1.2 Другий спосіб множення:

1.2.1 Вираз Z = XY = Yx12-1 + Yx22-2 +…+ Yxi2-i +…+ Yxn2-n можна подати у вигляді:

Z = ((…((0+Y2-nxn)+Y2-n+1xn-1) +…+ Y2-1x1

Очевидно, що процес множення може бути зведений до n-кратного виконання

циклу:

Zi = Zi-1 + Yixn-i+1, Yi = 2Yi-1, з початковими значеннями i = 1, Z0 = 0, Y0 = Y2-n.

У розглянутому способі множення здійснюється з молодших розрядів, множене

зсувається вліво, а сума часткових добутків залишається нерухомою.

1.2.2 Операційна схема:

2n-1 RG1 0 n-1 RG2**→** 0

**+**

2n-1 RG3**←** 0

Рис. 1.6 Операційна схема ІІ способу множення

1.2.3 Змістовний мікроалгоритм:

Початок

RG1:=0

RG2:=X RG3:=Y

0

RG2[0]

1

RG1:=RG1+

RG3

RG2:=0.r(RG2)

RG3:=l(RG3).0

0

RG2=0

1

Кінець

Рис. 1.7 Мікроалгоритм множення ІІ способом

1.2.4 Таблиця станів регістрів:

Табл. 1.2 – Таблиця станів регістрів при множенні ІІ способом

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG3 🡨 | RG2 🡪 |
| П.С. | 000000000000000000000000000000 | 000000000000000101000010001001 | 10010001100011**0** |
| 1 | 000000000000000000000000000000 | 000000000000001010000100010010 | 01001000110001**1** |
| 2 | +000000000000001010000110001010  000000000000001010000110001010 | 000000000000010100001000100100 | 00100100011000**1** |
| 3 | 000000000000010100001000100100 | 000000000000101000010001001000 | 00010010001100**0** |
| 4 | +000000000000101000010001001000  000000000000111100011001101100 | 000000000001010000100010010000 | 00001001000110**0** |
| 5 | 000000000000111100011001101100 | 000000000010100001000100100000 | 00000100100011**0** |
| 6 | 000000000000111100011001101100 | 000000000101000010001001000000 | 00000010010001**1** |
| 7 | 000000000000111100011001101100 | 000000001010000100010010000000 | 00000001001000**1** |
| 8 | +000000001010000100010010000000  000000001011000000101011101100 | 000000010100001000100100000000 | 00000000100100**0** |
| 9 | 000000001011000000101011101100 | 000000101000010001001000000000 | 00000000010010**0** |
| 10 | 000000001011000000101011101100 | 000001010000100010010000000000 | 00000000001001**0** |
| 11 | 000000001011000000101011101100 | 000010100001000100100000000000 | 00000000000100**1** |
| 12 | +000010100001000100100000000000  000010101100000101001011101100 | 000101000010001001000000000000 | 00000000000010**0** |
| 13 | 000010101100000101001011101100 | 001010000100010010000000000000 | 00000000000001**0** |
| 14 | 000010101100000101001011101100 | 010100001000100100000000000000 | 00000000000000**1** |
| 15 | +010100001000100100000000000000  **010110110100101001001011101100** | 101000010001001000000000000000 | **000000000000000** |

1.2.5 Функціональна схема:

X2

W

СLR

1

2n-1 RG1 0 X1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| n-1 RG2→ 0 | | | |
|  | | | |

2n+1

SM W SR

2n-1 0 2n-1 0 **X**

W

SL

|  |  |
| --- | --- |
|  | 2n+1 |
| 2n-1 **←**RG3 **0** | |
|  | |

**Y**

Рис. 1.8 – Функціональна схема з відображенням управляючих сигналів

1.2.6 Закодований мікроалгоритм:

Початок **Z1**

W2; CLR1; W3; **Z2**

0

X1

1

W1 **Z3**

SL3; SL2; **Z4**

0

X2

1

Кінець **Z5**

Рис. 1.9 – Закодований мікроалгоритм для множення ІІ способом

1.2.7 Граф управляючого автомата Мура:

**000**

**Z1 -**

**-**

**Z2 W2; CLR1; W3**

**001**

**Z5**

**- X2 X1 X1**

**111**

**Z4 -**

**SL3; SR2**

**Z3 W1**

**010**

**X1, X2**

**011**

**X1, X2**

Рис. 1.10 – Граф автомата регулювання управляючих сигналів

1.2.8 Обробка порядку результату:

Нормалізація мантиси.

MZ= ,010110110100101 🡨 ,

,10110110100101

Знак мантиси:

1.2.9 Запис результату в пам'ять

**Зн.Р P=+1210 Зн.М M**

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

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

1.3 Третій спосіб множення:

1.3.1 Вираз Z = XY = Yx12-1 + Yx22-2 +…+ Yxi2-i +…+ Yxn2-n можна подати у вигляді:

Z = ((…((0+Y2-nx1)2+Y2-nx2)2 +…+ Y2-nxi)2 +…+ Y2-nxn

Суму часткових добутків у і-му циклі можна одержати за виразом:

Zi = 2Zi-1 + Y2-nxi, з початковими значеннями i = 1, Z0 = 0.

У розглянутому способі множення здійснюється зі старших розрядів множника, сума

часткових добутків зсувається вліво, а множене нерухоме.

1.3.2 Операційна схема:

DL

n-1 **←** RG1 0 n RG1**←** 0

**+**

n-1 RG3 0

q СТ 0

Рис. 1.11 – Операційна схема пристрою для множення ІІІ способом

1.3.3 Змістовний мікроалгоритм:

Початок

RG1:=0;

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

0

RG2[n-1]

1

RG2.RG1:=

RG2.RG1+RG3

RG1:=l(RG1).0

RG2:=l(RG2).RG1[n] CT:=CT-1

0

CT=0

1

Кінець

Рис. 1.12 – Мікроалгоритм множення ІІІ способом

1.3.4 Таблиця станів регістрів:

Табл. 1.3 Таблиця станів регістрів при множенні ІІІ способом

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG1 | RG3 | RG2 | СТ |
| П.С. | 0000000000000000000000000000000 | 101000010001001 | 100100010001100 | 1111 |
| 1 | 000000000000001010000100010010 |  | 001000100011000 | 1110 |
| 2 | 000000000000010100001000100100 |  | 010001000110000 | 1101 |
| 3 | 000000000000101000010001001000 |  | 100010001100000 | 1100 |
| 4  🡨 | +000000000000000101000010001001  000000000000101101010011010001  000000000001011010100110100010 |  | 000100011000000 | 1011 |
| 5 | 000000000010110101001101000100 |  | 001000110000000 | 1010 |
| 6 | 000000000101101010011010001000 |  | 010001100000000 | 1001 |
| 7 | 000000001011010100110100010000 |  | 100011000000000 | 1000 |
| 8  🡨 | +000000000000000101000010001001  000000001011011001110110011001  000000010110110011101100110010 |  | 000110000000000 | 0111 |
| 9 | 000000101101100111011001100100 |  | 001100000000000 | 0110 |
| 10 | 000001011011001110110011001000 |  | 011000000000000 | 0101 |
| 11 | 000010110110011101100110010000 |  | 110000000000000 | 0100 |
| 12  🡨 | +000000000000000101000010001001  000010110110100010101000011001  000101101101000101010000110010 |  | 100000000000000 | 0011 |
| 13  🡨 | +000000000000000101000010001001  000101101101001010010010111011  001011011010010100100101110110 |  | 000000000000000 | 0010 |
| 14 | 010110110100101001001011101100 |  | 000000000000000 | 0001 |
| 15 | **,** **101101101001010010010111011000** |  | 000000000000000 | **0000** |

1.3.5 Функціональна схема:

X1

|  |  |  |
| --- | --- | --- |
|  | ... |  |
| q CT **0** | | |

n-1 **←**RG2 **0**

W

n

SL V

MX

W SL

CLR

n **←**RG1 **0**

n+1

**X**

SM

2n-1 0 2n-1 0

2n+1 n

X2

1 n-1 RG3 **0**

W

**Y**

Рис. 1.13 Функціональна схема множення ІІІ способом

1.3.6 Закодований мікроалгоритм:

Початок **Z1**

CLR1; WCT; **Z2**

W3; V; W2;

X1 0

1

W1; W2; **Z3**

SL2; SL1; DEC; **Z4**

0

X2

1

Кінець **Z1**

Рис. 1.14 – Закодований мікроалгоритм для множення ІІІ способом

1.3.7 Граф управляючого автомата Мура:

**000**

**Z1 -**

**-**

**Z2 W2; CLR1; W3; WCT; V**

**001**

**Z5**

**- X2 X1 X1**

**111**

**Z4 -**

**SL3; SL2;**

**DEC;**

**Z3 W1; W2**

**010**

**X1, X2**

**011**

**X1, X2**

Рис. 1.15 – Граф управляючого автомата

1.3.8 Обробка порядку результату:

.

Нормалізація мантиси.

MZ= , 010110110100101 🡨 , ,

, 10110110100101 .

Знак мантиси: .

1.3.9 Запис результату в пам'ять

Зн.Р P=+1210 Зн.М M

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

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

1.4 **Множення ІV-им способом:**

1.4.1 Вираз Z = XY = Yx12-1 + Yx22-2 +…+ Yxi2-i +…+ Yxn2-n можна подати у вигляді:

Z = ((…((0+Y2-1x1)+Y2-2x2) +…+ Y2-ixi) +…+ Y2-nxn

У цьому випадку процес множення може бути зведений до n-кратного виконання

циклу

Zi = Zi-1 + Yi-1xi, Yi = Yi-12-1, з початковими значеннями i = 1, Y0 = Y2-1, Z0 = 0.

У розглянутому способі множення здійснюється зі старших розрядів множника, сума

часткових добутків залишається нерухомою, а множене зсувається вправо.

1.4.2 Операційна схема:

2n-1 RG1 0 n-1 RG2**←** 0

**+**

2n-1 RG3**→** 0

Рис. 1.16 – Операційна схема пристрою множення ІV способом

1.4.3 Змістовний мікроалгоритм:

Початок

RG1:=0;

RG2:=X; RG3:=Y;

0

RG2[n-1]

1

RG1:=RG1+RG3

RG3:=0.r(RG3)

RG2:=l(RG2).0

RG2=0 0

1

Кінець

Рис. 1.17 – Мікроалгоритм множення ІV способом

1.4.4 Таблиця станів регістрів:

Табл. 1.4 – Таблиця станів регістрів для множення ІV способом

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG3 | RG2 |
| П.С. | 000000000000000000000000000000 | 010100001000100100000000000000 | 100100010001100 |
| 1 | 010100001000100100000000000000 | 001010000100010010000000000000 | 001000100011000 |
| 2 | 010100001000100100000000000000 | 000101000010001001000000000000 | 010001000110000 |
| 3 | 010100001000100100000000000000 | 000010100001000100100000000000 | 100010001100000 |
| 4 | +000010100001000100100000000000  010110101001101000100000000000 | 000001010000100010010000000000 | 000100011000000 |
| 5 | 010110101001101000100000000000 | 000000101000010001001000000000 | 001000110000000 |
| 6 | 010110101001101000100000000000 | 000000010100001000100100000000 | 010001100000000 |
| 7 | 010110101001101000100000000000 | 000000001010000100010010000000 | 100011000000000 |
| 8 | +000000001010000100010010000000  010110110011101100110010000000 | 000000000101000010001001000000 | 000110000000000 |
| 9 | +010110110011101100110010000000 | 000000000010100001000100100000 | 001100000000000 |
| 10 | 010110110011101100110010000000 | 000000000001010000100010010000 | 011000000000000 |
| 11 | 010110110011101100110010000000 | 000000000000101000010001001000 | 110000000000000 |
| 12 | +000000000000101000010001001000  010110110100010101000011001000 | 000000000000010100001000100100 | 100000000000000 |
| 13 | +000000000000010100001000100100  **,010110110100101001001011101100** | 000000000000001010000100010010 | 000000000000000 |

1.4.5 Функціональна схема:

X2

W

СLR

1

2n RG1 0 X1

|  |  |  |  |
| --- | --- | --- | --- |
|  | ... |  |  |
| n-1 **←**RG2 0 | | | |
|  | | | |

2n+1

SM W SL

2n 0 2n-1 0 **X**

W

SR

|  |  |
| --- | --- |
|  | 2n |
| 2n-1 RG3→ **0** | |
|  | |

**Y**

Рис. 1.18 Функціональна схема з відображенням управляючих сигналів

1.4.6 Закодований мікроалгоритм:

Початок **Z1**

W2; W3; CLR1; **Z2**

0

X1

1

W1 **Z3**

SR3; SL2; **Z4**

0

X2

1

Кінець **Z4**

Рис. 1.19 – Закодований мікроалгоритм для множення ІV способом

1.4.7 Граф управляючого автомата Мура:

000

**Z1** -

**-**

**Z2 W2; CLR1; W3**

001

**Z5**

**-** X2 X1 X1

111

**Z4** -

**SL2; SR3**

**Z3 W1**

010

X1, X2

011

X1, X2

Рис. 1.20 – Граф автомата мура для множення ІV способом

1.4.8 Обробка порядку результату:

.

Нормалізація мантиси.

MZ= , 010110110100101 🡨 , ;

,10110110100101 .

Знак мантиси: .

1.4.9 Запис результату в пам'ять:

Зн.Р P=+1210 Зн.М M

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

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

**2 Операція ділення чисел**

**2.1 Ділення першим способом**

2.1.1 Під час ділення за першим способом 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 записане ділене X). Ділене Y знаходиться в регістрі RG1. Результат формується в регістрі RG3 за (n+1) циклів. Алгоритм ділення зводиться до

виконання наступних дій:

1) Одержати різницю R0 = X-Y. Якщо R0≥0, то цифра частки Z0, що має вагу

20, дорівнює 1, а за R0<0 – дорівнює 0. Різниця R0 є залишком.

2) Подвоїти залишок (тобто одержати значення 2Ri).

3) За 2Ri<0 додати Y, в зворотному випадку, якщо 2R0 ≥ 0, відняти Y. Якщо

знову отриманий залишок Ri+1 ≥ 0, то Zi+1 = 1, інакше Zi+1 = 0.

4) Повторити дії, описані в пунктах 2 та 3, (n-1) раз.

2.1.2 Операційна схема:

n RGZ**← 0**

DL

n+1 RGX**← 0**

**+/-**

n+1 RGY **0**

Рис. 2.1 – Операційна схема ділення І способом

2.1.3 Функціональний мікроалгоритм:

Початок

RGZ:=0;

RGX:=00.X; RY:=00.Y;

RGZ:=l(RGZ).RGX[n+1];

RGX:=l(RGX).0

0 1

RGX[n+1]

RGX:=RGX+RGY+1 RGX:=RGX+RGY

0

RGZ[n]

1

Кінець

Рис. 2.2 Мікроалгоритм ділення І способом

2.1.4 Таблиця станів регістрів:

Табл. 2.1 – Таблиця станів регістрів при діленні І способом

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RG3 | RG2 | RG1 | МО |
| П.С. | 0000000000000000 | 00100100011000110 | 00101000010001001пк  11010111101110111дк | ПС |
| 1 | 0000000000000001 | 01001000100011000  +11010111101110111  00100000010001111 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 2 | 0000000000000011 | 01000000100011110  +11010111101110111  00011000010010101 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 3 | 0000000000000111 | 00110000100101010  +11010111101110111  00001000010100001 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 4 | 0000000000001111 | 00010000101000010  +11010111101110111  11101000010111001 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 5 | 0000000000011110 | 11010000101110010  +00101000010001001  11111000111111011 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 6 | 0000000000111100 | 11110001111110110  +00101000010001001  00011010001111111 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 7 | 0000000001111001 | 00110100011111110  +11010111101110111  00001100001110101 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 8 | 0000000011110011 | 00011000011101010  +11010111101110111  11110000001100001 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 9 | 0000000111100110 | 11100000011000010  +00101000010001001  00001000101001011 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 10 | 0000001111001101 | 00010001010010110  +11010111101110111  11101001000001101 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 11 | 0000011110011010 | 11010010000011010  +00101000010001001  11111010010100011 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 12 | 0000111100110100 | 11110100101000110  +00101000010001001  00011100111001111 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 13 | 0001111001101001 | 00111001110011110  +11010111101110111  00010001100010101 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 14 | 0011110011010011 | 00100011000101010  +11010111101110111  11111010110100001 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 15 | 0111100110100110 | 11110101101000010  +00101000010001001  00011101111001011 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |
| 16 | **1,111001101001101** | 00,111011110010110  +11,010111101110111  00,010011100001101 |  | 🡨 RG2🡨RG3  RG2:=RG2-RG1 |

2.1.5 Функціональна схема:

X1 X2

CLR SL

n **←**RGZ **0**

DL

n **←**RGX **0**

W

SL MX

**X**

SM

S

n+1 0 n+1 0

2n+1

W

=1

...

n+1 RGY **0**

n

**Y**

Рис. 2.3 Функціональна схема для ділення І способом

2.1.6 Закодований мікроалгоритм:

Початок **Z1**

WX; V; WY; CLRZ; **Z2**

WZ;

SLZ; SLX; **Z3**

1 0

X1

WX; SRY; SLZ; **Z4**

S; WX; SRY;

SLZ; **Z5**

0

X2

1

Кінець **Z6**

Рис. 2.4 – Закодований мікроалгоритм для ділення І способом

2.1.7 Граф управляючого автомата Мура:

100

**Z1** -

**-**

**Z2**

**WX; V; WY; CLRZ;**

000

**Z6**

**-**

111

X2

X2

**Z4 S; WX;**

110

**Z5 WX;**

X1

X2

X2

X1

**Z3 SLX; SLZ;**

010

011

Рис. 2.5 Граф автомата для ділення І способом

2.1.8 Обробка порядку результату:

.

Нормалізація мантиси не потрібна.

MZ= ,111001101001101.

Знак мантиси: .

2.1.9 Запис результату Z=X/Y в пам'ять:

Зн.Р P=+310 Зн.М M

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

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

2.2 **Ділення ІІ-им способом**

2.2.1 Під час ділення другим способом, який називається діленням із зсувом дільника, збільшується розрядність регістрів RG1, RG3 і суматора SM. В цьому випадку процеси додавання-віднімання і зсуву можуть бути сполучені в часі. Цифра результату формується на виході суматора SM(p).

Алгоритм ділення зводиться до виконання наступних дій:

1) Одержати різницю R0 = X-Y. Якщо R0≥0, то цифра частки Z0, що має вагу

20, дорівнює 1, а за R0<0 – дорівнює 0. Різниця R0 є залишком.

2) Зменшити вдвічі дільник.

3) За Ri<0 додати Y, в зворотному випадку, якщо R0 ≥ 0, відняти Y. Якщо знову отриманий залишок Ri+1 ≥ 0, то Zi+1 = 1, інакше Zi+1 = 0.

4) Повторити дії, описані в пунктах 2 та 3, (n-1) раз.

2.2.2 Операційна схема:

2n RGX 0

p n-1 RGZ**←** 0

**+/-**

2n RGY**→** 0

Рис. 2.6 Операційна схема ділення ІІ способом

2.2.3 Функціональний мікроалгоритм:

Початок

RGX:=0.X;

RGY:= 00.Y; RGZ:=0..01;

1 0

RGX[2n]

RGX:=RGX+RGY

RGY:=0.r(RGY) RGZ:=l(RGZ).P

RGX:=RGX+RGY+1

RGY:=0.r(RGY) RGZ:=l(RGZ).P

0

RGZ[n-1]

1

Кінець

Рис. 2.7 – Мікроалгоритм ділення ІІ способом

2.2.4 Таблиця станів регістрів:

Табл. 2.2 Таблиця станів регістрів ділення ІІ способом

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG3 | RG2 | RG1 |
| П.С. | 0000000000000001 | 0,10010001000110000000000000000 | 00,1010000100010010000000000000пк  11,0101111011101110000000000000дк |
| 1 | 0000000000000011 | +110101111011101110000000000000  001000000100011110000000000000 | 000101000010001001000000000000 |
| 2 | 0000000000000111 | +111010111101110111000000000000  000011000010010101000000000000 | 000010100001000100100000000000 |
| 3 | 0000000000001111 | +111101011110111011100000000000  000000100001010000100000000000 | 000001010000100010010000000000 |
| 4 | 0000000000011110 | +111110101111011101110000000000  111111010000101110010000000000 | 000000101000010001001000000000 |
| 5 | 0000000000111100 | +000000101000010001001000000000  111111111000111111011000000000 | 000000010100001000100100000000 |
| 6 | 0000000001111001 | +000000010100001000100100000000  000000001101000111111100000000 | 000000001010000100010010000000 |
| 7 | 0000000011110011 | +111111110101111011101110000000  000000000011000011101010000000 | 000000000101000010001001000000 |
| 8 | 0000000111100110 | +111111111010111101110111000000  111111111110000001100001000000 | 000000000010100001000100100000 |
| 9 | 0000001111001101 | +000000000010100001000100100000  000000000000100010100101100000 | 000000000001010000100010010000 |
| 10 | 0000011110011010 | +111111111110101111011101110000  111111111111010010000011010000 | 00000000000010100001000100100 |
| 11 | 0000111100110100 | +000000000000101000010001001000  111111111111111010010100011000 | 000000000000010100001000100100 |
| 12 | 0001111001101001 | +000000000000010100001000100100  000000000000001110011100111100 | 000000000000001010000100010010 |
| 13 | 0011110011010011 | +111111111111110101111011101110  000000000000000100011000101010 | 000000000000000101000010001001 |
| 14 | 0111100110100110 | +111111111111111010111101110111  111111111111111111010110100001 | 000000000000000010100001000100 |
| 15 | **1,111001101001101** | +000000000000000010100001000100  000000000000000001110111100101 | 000000000000000001010000100010 |

2.2.5 Функціональна схема:

X1 X2

CLR SL

n-1 **←**RGZ **0** P

1

2n RGX **0**

W

MX V

**X**

SM

S

2n 0 2n 0

2n+1 =1

n

W

SL **Y**

|  |  |  |
| --- | --- | --- |
|  | ... |  |
| 2n RGY **0** | | |
|  | | |

Рис. 2.8 – Функціональна схема з відображенням управляючих сигналів

2.2.6 Закодований мікроалгоритм:

Початок **Z1**

WX; V; WY; CLRZ; **Z2**

WZ;

1 0

X1

WX; SRY; SLZ; **Z3**

S; WX; SRY; **Z4**

SLZ;

0

X2

1

Кінець **Z5**

Рис. 2.9 Закодований мікроалгоритм для ділення ІІ способом

2.2.7 Граф управляючого автомата Мура:

**000**

**Z1 -**

**-**

**Z2**

**WX; V; WY; CLRZ; WZ;**

**001**

**X1**

**Z1 X2 X1**

**-**

**X2**

**111**

**Z4 S; WX; SRY; SLZ;**

**X1, X2**

**X1, X2**

**Z3 WX; SRY; SLZ;**

**010**

**X1, X2**

**011**

**X1, X2**

Рис. 2.10 Граф управляючого автомата для ділення ІІ способом

2.2.8 Обробка порядку результату:

Нормалізація мантиси не потрібна.

MZ= ,,111001101001101

Знак мантиси:

2.2.9 Запис результату Z=Y/X в пам'ять:

Зн.Р P=+310 Зн.М M

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

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

3 **Додавання чисел**

1) Px>Py, отже Pz=Px=0.0001000;

Λ=11

2) Вирівнюємо порядки

|  |  |  |
| --- | --- | --- |
|  |  | Мікрооперація |
| 0.101000010001001 | 11 | П.С. |
| 0.010100001000100 | 10 | 🡪 |
| 0.001010000100010 | 01 | 🡪 |
| 0.000101000010001 | 00 | 🡪 |

3) Перетворимо в ДК та підсумуванням:

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

4) Перетворимо результат назад в ПК:

Решту 9 пунктів виконаємо для нормалізації:

3.1 В машинах підсумування і віднімання кодів виконується на суматорах.

Віднімання замінюється на додавання за правилом: один операнд змінює знак.

Прямий код не застосовується для підсумування.

В арифметичному пристрої знаковий розряд подвоюється, щоб не було переповнення розрядної сітки.

Перед підсумування необхідно виконати вирівнювання порядків операндів. Після цього здійснюється підсумування мантис операндів на суматорі за участі суматора переносу.

3.2 Операційна схема:

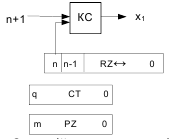


Рис.3.1 – Операційна схема нормалізації при додаванні

3.3 Функціональний мікроалгоритм:

Початок

RZ:=Z; PZ:=PZ; CT:=n;

1

RZ:=0.r(RZ) PZ:=PZ+1;

0

X1

1 0

RZ[N-1]

RZ:=l(RZ).0;

PZ:=PZ-1; CT:=CT-1;

1 0

CT=0

RZ:=0..01;

RZ:=1..1;

Кінець

Рис. 3.2 – Мікроалгоритм додавання

3.4 Таблиця станів регістрів:

Табл. 3.1 Таблиця станів регістрів при додаванні

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ такту** | **RGPZ** | **RGZ** | **ЛПН(L)** | **ППН(R)** | **СT** | **Мікрооперація** |
| **ПС** | 001000 |  | 0 | 1 | 011 |  |
| **1**  **2**  **3** | 000111 | **0** | 0 | 0 | 010  001  000 | Z’0 Z0:=  RGZ:=l(RGZ).0  RGPZ:=RGPZ-1  CT:=CT-1 |

3.5 Функціональна схема:

n+1

КС x1 X3

х2 1

|  |  |  |
| --- | --- | --- |
|  | ... |  |
| q CT 0 | | |
|  | | |

n n-1 RZ**↔** 0

W

SL

W

dec

SR MX

0..01 Z

V

inc dec W

S

n

m PZ 0

PZ

Рис. 3.3 – Функціональна схема з відображенням управляючих сигналів для додавання

3.6 Закодований мікроалгоритм:

Початок

WRZ; WPZ; WCT;

1 0

X1

SRRZ; incPZ; 1 0

X2

SLRZ;

decPZ;

decCT

1 0

X3

V; WRZ; S

Кінець

Рис. 3.4 – Закодований мікроалгоритм для додавання

3.7 Граф управляючого автомата Мура:

-

**Z1**

**-**

**Z2 WRZ; WPZ;**- **WCT;**

x1

x1 x2

**Z6**

**Z3 SRRZ; incPZ;**

-

x1 x2 - **-**

**Z4 SLRZ; decPZ; decCT;**

x3 x2

x3

-

**Z5 V; WRZ; S;**

Рис. 3.5 Граф автомата управління пристроєм додавання

3.8 Обробка порядку результату входить в саму схему.

3.9Запис результату Z=Y+X в пам'ять:

Зн.Р P=+710 Зн.М M

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

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

4 **Добування кореня**

4.1 Розглянемо метод обчислення функції  за послідовного введення коду аргументу X зі старших розрядів. Такий метод дозволяє виконувати процеси введення аргументу та обчислення функції у режимі суміщення.

Будемо вважати, що значення аргументу подане нормалізованим n–розрядним числом у позиційній системі числення з основою 2.

Процес обчислення квадратного кореня зводиться до послідовного виконання n

кроків. Перед початком обчислення X0 = Y0 = R0 = 0. На кожному i-му кроці (i є

(1,n)) цифру yi визначають відповідно системі:



Значення Yi та Ri – за формулами:

Yi = Yi-1 + yi2-i

Ri = 2Ri-1 + xi2-1 – Yi-1yi-yi22-i-1

При цьому формула обчислення Yi забезпечує автоматичне перетворення коду

результату з надлишкової системи числення в не надлишкову, якщо операцію

додавання виконувати в не надлишковій системі числення.

Процес обчислення квадратного кореня за запропонованим алгоритмом закінчується безпосередньо після введення останнього розряду коду аргументу.

Це дозволяє зменшити час обчислення в тих випадках, коли швидкість надходження чергових розрядів аргументу визначається зовнішніми стосовно операційного блоку факторами.

4.2 Операційна схема:

DL

n+1 RR**←←** 0 n-1 RX**←←** 0

**+ -**

n+1 RZ**←** 0

n CT 0

Рис. 4.1 Операційна схема пристрою для добування кореня

4.3 Функціональний мікроалгоритм:

Початок

RR:=0;

RX:=x; RZ:=0

0 1

RR(n+1)

RR:=RR+RZ.11 RR:=RR+RZ.11

RR:=l2(RR).RX[n-1,

n-2] RX:=l2(RX).00 RZ:=l(RZ).RR[n+1] CT:=CT-1

0

CT=0

1

Кінець

Рис. 4.2 Мікроалгоритм добування кореня

4.4 Таблиця станів регістрів:

Табл. 4.1 Таблиця станів регістрів пристрою для добування кореня

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № ц. | RZ | RR | RX | CT |
| П.С.  Пзсув | 000000000000000 | 00000000000000000  **0**0000000000000010 | 100100011000110  010001100011000 | 1111 |
| 1 | 000000000000001 | +11111111111111111  00000000000000001  **0**0000000000000101 | 000110001100000 | 1110 |
| 2 | 000000000000011 | +11111111111111011  00000000000000000  **0**0000000000000000 | 011000110000000 | 1101 |
| 3 | 000000000000110 | +11111111111110011  11111111111110011  **1**1111111111001101 | 100011000000000 | 1100 |
| 4 | 000000000001100 | +00000000000011011  11111111111101000  **1**1111111110100010 | 001100000000000 | 1011 |
| 5 | 000000000011000 | +00000000000110011  11111111111010101  **1**1111111101010100 | 110000000000000 | 1010 |
| 6 | 000000000110000 | +00000000001100011  11111111110110111  **1**1111111011011111 | 000000000000000 | 1001 |
| 7 | 000000001100000 | +00000000011000011  11111111110100010  **1**1111111010001000 | 000000000000000 | 1000 |
| 8 | 000000011000001 | +00000000110000011  00000000000001011  **0**0000000000101100 | 000000000000000 | 0111 |
| 9 | 000000110000010 | +11111110011111011  11111110100100111  **1**1111010010011100 | 000000000000000 | 0110 |
| 10 | 000001100000100 | +00000011000001011  11111101010100111  **1**1110101010011100 | 000000000000000 | 0101 |
| 11 | 000011000001000 | +00000110000010011  11111011011000011  **1**1101101100001100 | 000000000000000 | 0100 |
| 12 | 000110000010000 | +00001100000100011  11111001100101111  **1**1100110010111100 | 000000000000000 | 0011 |
| 13 | 001100000100000 | +00011000001000011  11111110011111111  **1**1111001111111100 | 000000000000000 | 0010 |
| 14 | 011000001000001 | +00110000010000011  00101010001111111  **1**0101000111111100 | 000000000000000 | 0001 |
| 15 | 110000010000011 | +01100000100000111  00001001100000011 | 000000000000000 | 0000 |

4.5 Функціональна схема:

|  |  |
| --- | --- |
|  |  |
| q CT 0 | |
|  | |

CLR SL W

n+1 RR**← 0**

SM

n+1 0 n-1 0

n-1 RX**← 0**

W

SL

**X**

S

x2

1

=1

CLR SL

n-1 RZ**← 0**

W

dec

Рис. 4.3 Функціональна схема добування кореня

4.6 Закодований мікроалгоритм:

Початок **Z1**

WX; WCT;

CLRR; CLRZ; **Z2**

SLR; SLX;

0 1

X1

S; WR **Z3**

WR; **Z4**

SLR; SLX; SLZ; **Z5**

decCT

0

X2

1

Кінець **Z6**

Рис. 4.4 Закодований мікроалгоритм добування кореня.

4.7Граф управляючого автомата Мура:

**Z2**

**WX; WCT; CLRR;CLRZ;**

- **SLR; SLX;**

X1  **Z3 S; WR;**

**Z1** X1

**-**

X1, X2

X1, X2

**Z6**

- **-**

X2

**Z4**

**WR;** -

**Z5 SLR; SLX; SLZ; decCT;**

Рис. 4.5 Граф Автомата регулювання пристрою добування кореня

4.8 Обробка порядку результату:

PZ=PX:2=8:2=410=1002.

4.9 Запис результату Z= в пам'ять:

Зн.Р P=+410 Зн.М M

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

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

**5. Синтез управляючого автомату для операційного пристрою множення третім способом.**

x3x2x1 =011=3 – операція множення третім способом.

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

За закодованим мікроалгоритмом (Рис. 1.14) складемо таблицю:

*Таблиця 5.1 Таблиця кодування сигналів*

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

**5.2 Мікроалгоритм в термінах управляючого автомата**

Зробимо автомат Мура циклічним задля зменшення кількості вершин.

Початок

Z1

Y1

Z2

X1

1

Y2

Z3

Z4

Y3

X2

1

Кінець

Z1

*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*

**5.3 Структурна таблиця автомата**

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

*Таблиця 5.2-Структурна таблиця автомата*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Перехід | Q2Q1 | Q2Q1 | x1x2 | y1y2y3 | J2 | K2 | J1 | K1 |
| *z1z2* | 00 | 01 | -- | 000 | 0 | - | 1 | - |
| *z2 z3* | 01 | 11 | 1- | 100 | 1 | - | - | 0 |
| *z2 z4* | 01 | 10 | 0- | 100 | 1 | - | - | 1 |
| *z3 z4* | 11 | 10 | -- | 010 | - | 0 | - | 1 |
| *z4 z3* | 10 | 11 | 10 | 001 | - | 0 | 1 | - |
| *z4 z1* | 10 | 00 | -1 | 001 | - | 1 | 0 | - |
| *z4 z4* | 10 | 10 | 00 | 001 | - | 0 | 0 | - |

**5.4 Синтез функцій виходів і переходів**

JK-тригер:

0

1

0

0

Y3

1

0

0

0

Y2

0

0

1

0

Q2

Q1

Y1

1

1

-

-

1

1

-

-

0

0

-

-

1

1

-

-

Q2

Q1

X2

X1

K1

-

-

0

0

-

-

0

1

-

-

1

1

-

-

1

1

Q2

Q1

X2

X1

J1

0

0

1

0

0

0

-

1

0

-

-

-

-

-

-

-

-

Q2

Q1

X2

X1

K2

-

-

-

-

-

-

-

-

1

1

0

0

1

1

0

0

Q2

Q1

X2

X1

J2

*Рисунок 5.3- Діаграми Вейча*

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

**підключені до входів операційного автомата)**

­­

K2

R

K

C

S

J

G

“1”

D2

R

K

C

S

J

J1

&

&

&

Y1

Y2

Y3

“1”

***Y1***

***Y2***

***Y3***

***RG1***

***2n***

***0***

***RG2***

***0***

***n-1***

***n***

***X***

***Y1***

***Y3***

***2n+1***

***2n+1***

***2n+1***

***0***

***0***

***SM***

***x2***

***n+1***

***2n+1***

***n***

&

J1

1

**0**

***RG3***

K2

K1

&

***Y1***

***Y3***

***CT***

***s***

***n***

***Y***

***Y1***

***n-1***

***0***

1

*Рисунок 5.5- Функціональна схема пристрою*

**Висновки**

В даній роботі ми розглянули 8 основних операцій в машинних кодах: чотири способи множення, два способи ділення, додавання та добування кореня з . Для операції з номером множення ІІІ способом було побудовано управляючий автомат Мура на тригерах і елементах булевого базису.