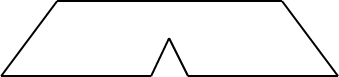
Потім виконують додавання мантис у доповнювальних кодах. Останній етап – нормалізація результату. Виконується за допомогою зсуву мантиси результату і коригування порядку результату. Порушення нормалізації можливо вліво і вправо, на 1 розряд вліво і на n розрядів вправо.

2.7.2) *Операційна схема:*



КС

m + 1 Rz 0

n + 1 RPz 0

q CT 1

L

R

m - кількість розрядів мантиси

n - кількість розрядів порядку

q = ]log2m[

S M

*Рисунок 2.31 – Операційна схема*

Виконаємо синтез комбінаційної схеми для визначення порушення нормалізації.

*Таблиця 2.19 – Визначення порушення нормалізації*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Розряди регістру Rz | | | 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:*

*Рисунок 2.32 – Змістовний мікроалгоритм*

0

1

1

0

так

ні

0

1

Початок

0

CT = 0

Кінець

Rz := Rz(m + 2).r(Rz);

RPz := RPz + 1;

R =

Z’0 = 0

CT := 15;

Rz := z;

Z’0Z0 :=

Rz := l(Rz).0;

RPz := RPz – 1;

CT := CT - 1;

2.7.4) *Таблиця станів регістрів (лічильника):*

Спочатку виконаємо порівняння порядків:

,

Потім вирівняємо їх, зсуваючи мантису Y на :

*Таблиця 2.20 – Таблиця зсуву мантиси на етапі порівняння порядків*

|  |  |  |
| --- | --- | --- |
|  |  | Мікрооперація |
| 0.101001110100001 | 11 | П.С. |
| 0.010100111010000 | 10 |  |
| 0.001010011101000 | 01 |  |
| 0.000101001110100 | 00 |  |

Додавання мантис у доповнювальних кодах:

11.011010001011100

+00.000101001110100

11.011111011010000

*Таблиця 2.21 – Таблиця станів регістрів*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | RPz | Rz | ЛПН(L) | ППН(R) | СT | Мікрооперація |
| П.С. | 001000 | 00. | 0 | 1 | 100 |  |
| 1 | 000111 | 00. | 0 | 0 | 011 | Z’0 Z0 :=  Rz := l(Rz).0  RPz:=RPz - 1  CT := CT - 1 |

Нормалізація не потрібна.

2.7.5) *Функціональна схема з відображенням управляючих сигналів:*

R

m + 1 m m - 1 Rz 0

n + 1 RPz 0

q CT 1

L

Z0



SL

SR

W1, W2

Inc

Dec

W

RCT

Dec

*Рисунок 2.33 – Функціональна схема*

2.7.6) *Закодований мікроалгоритм:*

*Таблиця 2.22 – Кодування мікрооперацій*

|  |  |  |
| --- | --- | --- |
| Мікрооперації | | Управляючі сигнали |
| CT := 15 | | W |
| Rz := Z | | W1 |
| Z’0Z0 := | | W2 |
| Rz := Rz(m + 2).r(Rz) | | SR |
| RPz := RPz + 1 | | Inc |
| Rz := l(Rz).0 | | SL |
| CT := CT – 1 | | Dec |
| RPz := RPz – 1 | | Dec |
|  | | |
| Логічна умова | Позначення | |
| Z’0 = 0 | X1 | |
| 0 | X2 | |
|  | X3 | |
|  | X4 | |

*Таблиця 2.23 – Кодування логічних умов*

*Таблиця 2.3 – Кодування логічних умов*

*Таблиця 2.3 – Кодування логічних умов*

*Таблиця 2.3 – Кодування логічних умов*

*Рисунок 2.34 – Кодований мікроалгоритм*

Z5

Z3

1

0

1

0

так

ні

0

1

Початок

W, W1

X2

SL, Dec, Dec

X4

Кінець

SR, Inc

X3

X1

W2

Z1

Z2

Z4

Z6

2.7.7) *Граф управляючого автомата Мура з кодами вершин:*

**

X2

Z2

W, W1

Z3

W2

Z4

SR, Inc

111

Z6

–

001

011

010

Z5

SL, Dec, Dec

110

–

X2

Z1

–

000

X4

–

*Рисунок 2.35 – Граф управляючого автомата*

2.7.8) *Обробка порядків:*

Порушення нормалізації немає, округлення виконане методом додавання 1 в молодший розряд.

2.7.9) *Форма запису нормалізованого результату з плаваючою комою в пам’ять:*

Запишемо результат додавання Z2 в розрядну сітку в прямому коді у вигляді ПОРЯДОК МАНТИСА:

Z2 : **0.** 0 0 0 1 0 0 0 **1,** 1 0 0 0 0 0 1 0 0 1 1 0 0 0 1

Знакові розряди виділені.

**2.8) Добування кореня Z =**

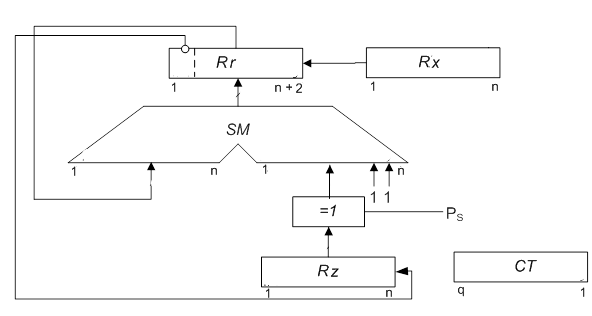
2.8.1) *Теоретичне обґрунтування способу*: розглянемо операцію , де X < 1.

Результат формується порозрядно, тому ;

Алгоритм має наступні етапи:

Цей спосіб також не є регулярним, тому його модифікують так, щоб остачу не відновлювати.

2.8.2) *Операційна схема:*

**

*Рисунок 2.36 – Операційна схема*

2.8.3) *Змістовний мікроалгоритм:*

2.8.4) *Таблиця станів регістрів (лічильника):*

так

ні

1

0

Початок

Rr := 0;

Rx := X2;

Rz := 0;

CT := 15;

Rr := l2(Rr).Rx[1, 2];

Rr[1]

Rr := Rz + Rr + 11;

Rr := l2(Rr).Rx[1, 2];

Rz := l(Rz).Rr[1];

CT := CT – 1;

Rx := l(Rx).0;

CT = 0

Кінець

*Рисунок 2.37– Змістовний мікроалгоритм*

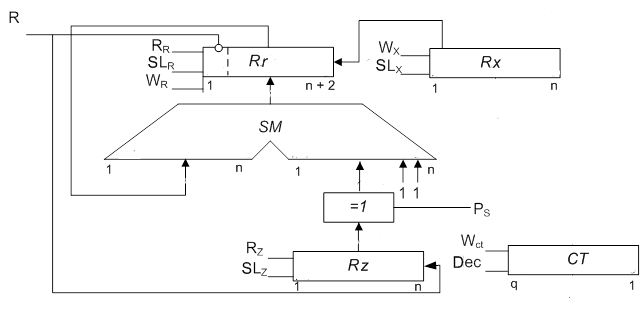
Rr := + Rr + 11;

*Таблиця 2.24 – Таблиця станів регістрів*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Rz | Rr | Rx | CT | Коментар |
| П.С. | 000000000000000 | 00000000000000000  **0**0000000000000010 | 10010111010010 | 1111 |  |
| 1 | 000000000000001 | 00000000000000010  +11111111111111111  00000000000000001  **0**0000000000000101 | 010010111010010 | 1110 | **+**  ⮘ |
| 2 | 000000000000011 | 00000000000000101  +11111111111111011  00000000000000000  **0**0000000000000001 | 001001011101001 | 1101 | **+**  ⮘ |
| 3 | 000000000000110 | 00000000000000001  +11111111111110011  11111111111110100  **1**1111111111010011 | 000100101110100 | 1100 | **+**  ⮘ |
| 4 | 000000000001100 | 11111111111010011  +00000000000011011  11111111111101110  **1**1111111110111001 | 000010010111010 | 1011 | **+**  ⮘ |
| 5 | 000000000011000 | 11111111110111001  +00000000000110011  11111111111101100  **1**1111111110110000 | 000001001011101 | 1010 | **+**  ⮘ |
| 6 | 000000000110001 | 11111111110110000  +00000000001100011  00000000000010011  **0**0000000001001110 | 000000100101110 | 1001 | **+**  ⮘ |
| 7 | 000000001100010 | 00000000001001110  +11111111100111011  11111111110001001  **1**1111111000100100 | 000000010010111 | 1000 | **+**  ⮘ |
| 8 | 000000011000100 | 11111111000100100  +00000000110001011  11111111110101111  **1**1111111010111100 | 000000001001011 | 0111 | **+**  ⮘ |
| 9 | 000000110001001 | 11111111010111100  +00000001100010011  00000000111001111  **0**0000011100111100 | 000000000100101 | 0110 | **+**  ⮘ |
| 10 | 000001100010011 | 00000011100111100  +11111100111011011  00000000100010111  **0**0000010001011100 | 000000000010010 | 0101 | **+**  ⮘ |
| 11 | 000011000100110 | 00000010001011100  +11111001110110011  11111100000001111  **1**1110000000111100 | 000000000001001 | 0100 | **+**  ⮘ |
| 12 | 000110001001100 | 11110000000111100  +00001100010011011  11111100011010111  **1**1110001101011100 | 000000000000100 | 0011 | **+**  ⮘ |
| 13 | 001100010011001 | 11110001101011100  +00011000100110011  00001010010001111  **0**0101001000111100 | 000000000000010 | 0010 | **+**  ⮘ |
| 14 | 011000100110010 | 00101001000111100  +11001110110011011  11110111111010111  **1**1011111101011100 | 000000000000001 | 0001 | **+**  ⮘ |
| 15 | **110001001100101** | 11011111101011100  +01100010011001011  01000010000100111  **0**0001000010011100 | 000000000000000 | 0000 | **+**  ⮘ |

Результат добування кореня виділено в таблиці.

2.8.5) *Функціональна схема з відображенням управляючих сигналів:*



*Рисунок 2.38 – Функціональна схема*

2.8.6) *Закодований мікроалгоритм:*

*Таблиця 2.25 – Кодування мікрооперацій*

|  |  |  |
| --- | --- | --- |
| Мікрооперації | | Управляючі сигнали |
| Rx := X2 | | WX |
| Rr := l2(Rr).Rx[1, L] | | RR |
| Rz := 0 | | RZ |
| CT := 15 | | WCT |
| Rr := Rr + Rz + 11 | | WR |
| Rz := l(Rz).Rr[1]; | | SLZ |
| Rr := l2(Rr).Rx[1, 2] | | SLR |
| CT := CT – 1 | | Dec |
| Rx := l(Rx).0 | | SLX |
| *Таблиця 2.26 – Кодування логічних умов* | | |
| Логічна умова | Позначення | |
| Rr[1] | X1 | |
| CT = 0 | X2 | |

*Рисунок 2.39 – Кодований мікроалгоритм*

Z5

Z3

Z4

так

ні

1

0

Початок

WX, RR, RZ, WCT

X1

WR

X2

Кінець

WR, PS

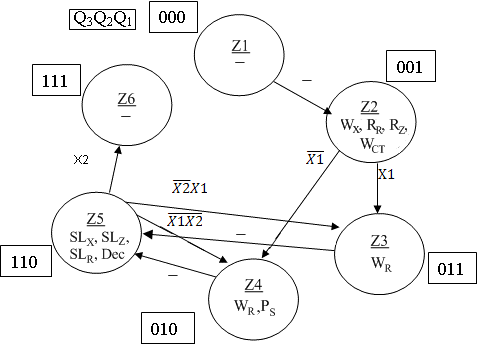
SLR, SLX, SLZ, Dec

Z1

Z2

Z6

2.8.7) *Граф управляючого автомата Мура з кодами вершин:*



2.8.8) *Обробка порядків:*

PZ = PX/2.

Отже порядок результату PZ = 1510/210 = 710 = 0.0000111

Порушення нормалізації немає.

2.8.9) *Форма запису нормалізованого результату з плаваючою комою в пам’ять:*

Запишемо результат добування кореня Z2 в розрядну сітку в прямому коді у вигляді ПОРЯДОК МАНТИСА:

Z2 : **0.** 0 0 0 0 1 1 1 **1,** 1 1 0 0 0 1 0 0 1 1 0 0 1 0 1

Знакові розряди виділені.

**3) Синтез управляючого автомату для операційного пристрою множення**

Оскільки x3x2x1 + 1 = 001, тому операція множення першим способом.

**3.1) Таблиця співвідношення управляючих входів операційного автомата і виходів управляючого автомата**

За закодованим мікроалгоритмом (Рис. 2.4) складемо таблицю:

*Таблиця 2.27– Кодування сигналів*

|  |  |
| --- | --- |
| Входи операційного автомата | Виходи управляючого автомата |
| RX, RY, RZ, RCT | Y1 |
| Sum | Y2 |
| SRX, SRZ, Dec | Y3 |

**3.2) Мікроалгоритм у термінах управляючого автомата**

Зробимо автомат Мура циклічним задля зменшення кількості вершин.

Z1

Z2

Z3

Z4

Z1

так

ні

1

0

Початок

Y1

X1

Y2

Y3

X2

Кінець

*Рисунок 2.41– Закодований мікроалгоритм*

Будуємо граф автомата Мура:

*Рисунок 2.42 – Граф управляючого автомата Мура*

, 

–

, 



X1



–

Q2Q1

00

01

11

10

**3.3) Структурна таблиця автомата**

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

*Таблиця 2.28 – Структурна таблиця автомата*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Перехід | Q2Q1 | Q2Q1 | x1x2 | y1y2y3 | D1 | D2 |
| *Z1→Z2* | 00 | 01 | −− | 000 | 0 | 0 |
| *Z2→Z3* | 01 | 11 | 1− | 100 | 0 | 1 |
| *Z2→Z4* | 01 | 10 | 0− | 100 | 0 | 1 |
| *Z3→Z4* | 11 | 10 | −− | 010 | 1 | 1 |
| *Z4→Z3* | 10 | 11 | 10 | 001 | 1 | 0 |
| *Z4→Z1* | 10 | 00 | −1 | 001 | 1 | 0 |
| *Z4→Z4* | 10 | 10 | 00 | 001 | 1 | 0 |

D-тригер:

**3.4) Синтез функцій виходів і переходів**

0

0

1

0

Q1

Q2

y3

0

1

0

0

Q1

Q2

y1

1

0

0

0

Q1

Q2

y2

1

1

1

0

1

1

-

1

11

-

-

-

-

-

-

-

-

Q1

Q2

x2

x1

D2

0

0

0

0

0

0

0

0

1

1

0

0

1

1

0

0

Q1

Q2

x2

x1

D1

*Рисунок 2.43 – Діаграми Вейча*

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

**підключені до входів операційного автомата)**

­­

D2

TT

D

C

S

R

Q2

“1”

G

D1

TT

D

C

S

R

Q1

“1”

R

&

&

&

1

Q1

x1

Q2

Q1

x2

D2

&

&

&

Y1

Y2

Y3

1

D1

&

*Рисунок 2.44 – Функціональна схема пристрою*

**Висновок:**

В даній роботі результати множення і ділення різними способами співпали, тому операції виконано правильно. Було удосконалено і систематизовано навички виконання арифметичних операцій в двійкових кодах над числами з плаваючою комою і побудови автоматів.

**Список літератури:**

1. Жабін В.І., Жуков І.А., Клименко І.А., Стіренко С.Г. – Прикладна теорія цифрових автоматів: навчальний посібник. – книжкове видавництво НАУ, 2007. – 364 с.
2. Самофалов К.Г., Романкевич А.М., Валуйский В.Н., Каневский Ю.С., Пиневич М.М. Прикладная теория цифровых автоматов. – К.: Вища школа, 1987. – 375 с.