

ЕЛЕМЕНТИ ТА ПРИСТРОЇ ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

УДК 681.325.5

СХЕМОТЕХНІЧНІ ОСНОВИ ПОБІТОВОГО ДОДАВАННЯ КОДІВ ЗОЛОТОЇ ПРОПОРЦІЇ

О.Д. Азаров, О.І. Черняк

Актуальність

Побітова конвеєрна обробка є окремим випадком порозрядної конвеєрної обробки, при якій досягається найменша кількість інформаційних зв'язків між розподіленими пристроями [1-4]. Коди золотої пропорції, запропоновані у [5], дозволяють виконувати усі арифметичні операції побітно, починаючи зі старших розрядів, забезпечуючи при цьому найменшу довжину перенесення у старші розряди. Тому актуальною є розробка схемотехнічних основ конвеєрного побітового додавання кодів золотої пропорції, як основи усіх арифметичних операцій над кодами чисел.

Аналіз останніх досліджень

У попередніх наукових працях авторами запропоновано загальний опис класу АМ-систем числення, що дозволяють виконувати порозрядну обробку зі старших розрядів [6]. Завдяки наявності у них адитивного співвідношення певного типу між вагами розрядів у цих системах числення можна виконувати адитивні перетворення над кодами чисел. Вони є узагальненням перенесення (AL-перетворення) і позичання (AR-перетворення) при додаванні і відніманні. За умовою виконання адитивні перетворення можуть бути елементарними (E-перетворення), універсальними (U-перетворення) та повними (F-перетворення). Окремим випадком АМ-систем числення є коди золотої пропорції, описані у [7,8]. Ці коди дозволяють виконувати конвеєрну побітову обробку, починаючи зі старших розрядів. Алгоритмічні основи побітової конвеєрної обробки кодів золотої пропорції описані у [9].

Мета статті

Метою статті є опис результатів, отриманих у процесі розробки схемотехнічних основ конвеєрного побітового додавання послідовних кодів золотої пропорції.

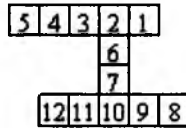
Постановка задач

Побітова обробка послідовних кодів золотої пропорції має певні особливості. При побітовому додаванні кодів золотої пропорції використовується код проміжного результату, що являє собою частину розрядів суми, отриманої на черговому такті, через які можливе розповсюдження перенесення у наступному такті виконання операції. Код проміжного результату формується за допомогою адитивних перетворень коду золотої пропорції. Таким чином, при розробці схемотехнічних основ конвеєрного побітового додавання кодів золотої пропорції постають такі задачі:

1. Розробка структурної та функціональної схем конвеєрного пристрою побітового додавання послідовних кодів золотої пропорції на основі адитивних перетворень.
2. Розробка функціональних схем окремих блоків.
3. Перевірка працездатності розроблених схем.

Розробка порозрядного суматора кодів золотої пропорції

Відповідно до розробленого алгоритму додавання суматор повинен на основі чергових розрядів доданків і попереднього проміжного результату формувати черговий розряд результату і черговий проміжний результат. Узагальнена структурна схема даного суматора подана на рисунку 1. Порозрядний суматор послідовних кодів золотої пропорції містить комбінаційний блок формування результатів, призначений для формування чергового розряду суми та коду S_i чергового проміжного результату, регістр R_g , призначений для зберігання коду S_{i-1} попереднього проміжного результату, вхід першого доданку x_i , вхід другого доданку y_i , вхід ТІ синхронізації, на який подаються тактові імпульси, вхід ПС встановлення у початковий стан та вихід z_i результату додавання, з якого подається черговий розряд суми. Регістр R_g має п'ять розрядів, що подаються на блок формування сигналів БФС. Крім того, на БФС подаються чергові розряди x_i та y_i доданків. Суматор виконує порозрядне додавання двох послідовних кодів золотої пропорції, починаючи зі старших розрядів. Формат даних суматора має вид:



де $\boxed{5\ 4\ 3\ 2\ 1}$ - код S_{i-1} регістра, що встановлюється на попередньому такті згідно з форматом вхідних даних і правилами арифметики чисел;

$\boxed{6}$ - тий розряд коду першого операнда,

$\boxed{7}$ - тий розряд коду другого операнда.

$\boxed{12\ 11\ 10\ 9\ 8}$ - код S_i регістра, що встановлюється для подальшого такту додавання згідно з форматом вхідних даних і правилами арифметики.

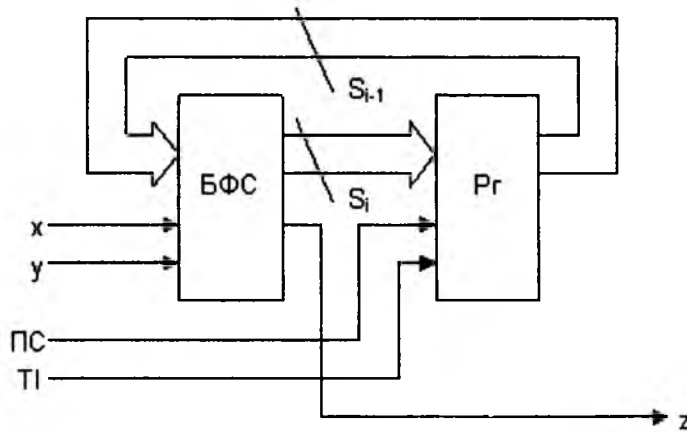


Рисунок 1 – Структурна схема порозрядного суматора послідовних кодів золоті пропорції

Суматор працює таким чином. Сигнал PC встановлює регістр у нульовий стан. Після зняття сигналу PC з надходженням кожного тактового імпульсу на входи доданків поступають послідовні коди даних, починаючи зі старших розрядів. У залежності від значень чергових розрядів доданків суматор з попереднього стану переходить в один з трьох нових станів. Оскільки результат додавання при значеннях чергових доданків 01 та 10 однаковий, то в подальшому будуть розглядатись тільки випадки 00, 01, 11. Код результату на виході суматора утворюється як послідовність окремих розрядів, що формуються на кожному такті додавання, починаючи з третього.

На рисунку 2 подано граф переходу станів суматора. Тут для спрощення рисунку переходи, що викликаються різними значеннями розрядів доданків зображені різними лініями:

- > - переходи, при $x=0$ і $y=0$;
- > - переходи, при $x=1$ і $y=0$, або $x=0$ і $y=1$;
- > - переходи, при $x=1$ і $y=1$;

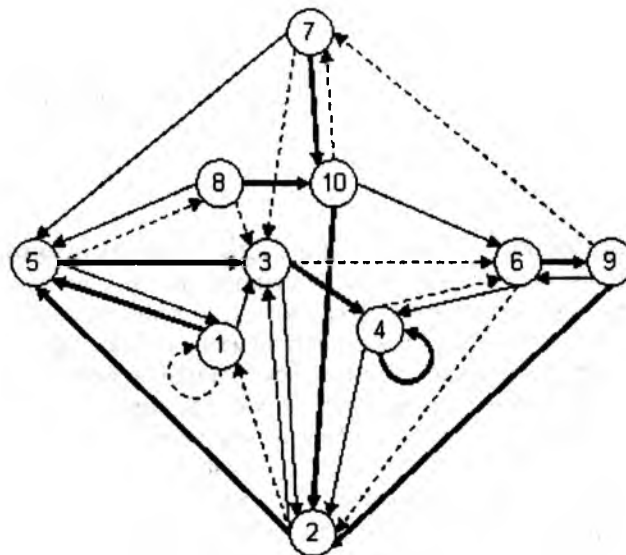


Рисунок 2 – Граф переходів суматора

Коди станів суматора такі: 1 – 00000, 2 – 00001, 3 – 00100, 4 – 10100, 5 – 01001, 6 – 01000, 7 – 00010, 8 – 10010, 9 – 00001, 10 – 10001. Аналіз рисунку 2 показує, що усі стани, крім п'ятого та шостого, можна поділити на дві симетричних множини, стани яких попарно відрізняються між собою тільки значенням старшого розряду коду. Це такі пари станів: (1 – 2), (3 – 4), (7 – 8), (9 – 10). Якщо старший розряд регістру мав одиничне значення, то встановлюється одиниця чергового розряду результату. І навпаки, нульове значення старшого розряду регістра для даних пар станів призводить до встановлення нульового значення чергового розряду результату. Зменшений граф переходів матиме лише шість станів, що можна позначити кодами: X0000, X0001, X0010, X0100, 01000, 01001, де X – будь-яке значення з множини {0,1}. Оскільки в останніх двох кодах значення чотирьох молодших розрядів повністю визначають відповідні стани, то для кодування усіх станів старший розряд можна відкинути. Значення старшого розряду проміжного результату повинно враховуватись лише при формуванні чергового розряду суми за допомогою функції АБО з сигналом, що виробляється при перенесенні внаслідок EAL-перетворення. Отже, коди станів будуть такими: 1 – 0000, 2 – 0001, 3 – 0010, 4 – 0100, 5 – 1000, 6 – 1001. На рисунку 3 подано оптимізований граф переходів суматора. Оптимізована структурна схема порозрядного суматора подана на рисунку 4.

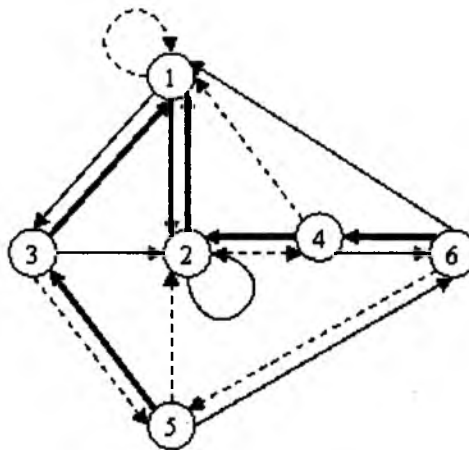


Рисунок 3 – Оптимізований граф переходів суматора

Дане рішення дозволяє зменшити кількість обладнання БФС суматора за рахунок того, що старший розряд проміжного результату не бере участі у формуванні коду наступного проміжного результату. Для спрощення схем, що реалізують переходи, введемо допоміжні сигнали A і B . Їх значення залежить від значень чергових розрядів доданків x_i та y_i , і задане за допомогою виразів:

$$A = XY;$$

$$B = (X + Y) \bar{A}.$$

Одиничне значення A означає, що сума чергових розрядів доданків дорівнює двом. Відповідно, одиничне значення B означає, що сума чергових розрядів доданків дорівнює одиниці. Якщо сума чергових розрядів доданків дорівнює нулю, то $A=0$ і $B=0$. Крім вказаних сигналів для реалізації логічних виразів, за якими виконуються переходи, введемо також сигнали, що утворюються за логічними виразами \bar{A} та $\bar{A}\bar{B}$. Одиничне значення сигналу $\bar{A}\bar{B}$ означає, що сума чергових розрядів доданків дорівнює нулю. Функціональна схема для визначення A , B та їх комбінацій, що будуть використовуватись далі, є схемою додавання чергових розрядів доданків і подана на рисунку 4.

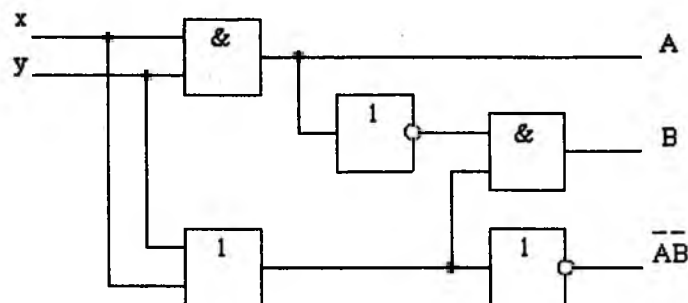


Рисунок 4 – Функціональна схема додавання чергових розрядів доданків

Для зберігання станів суматора оберемо регістр, побудований на динамічних D-тригерах. Виділемо в окремий блок операцію додавання розрядів. Таким чином, суматор складається з блоку БДР додавання чергових розрядів доданків, блоку БАП універсального адитивного перетворення з перенесенням в старші розряди, регістра Rg на динамічних тригерах та елемента АБО для формування чергового розряду суми. Для кодування шести станів регістра достатньо всього три розряди. Тому розглянемо два варіанти побудови суматора: 1 – з використанням повнорозрядного регістра; 2 – з використанням регістра зменшеної розрядності. Функціональна схема суматора з використанням повнорозрядного регістра подана на рис 5.

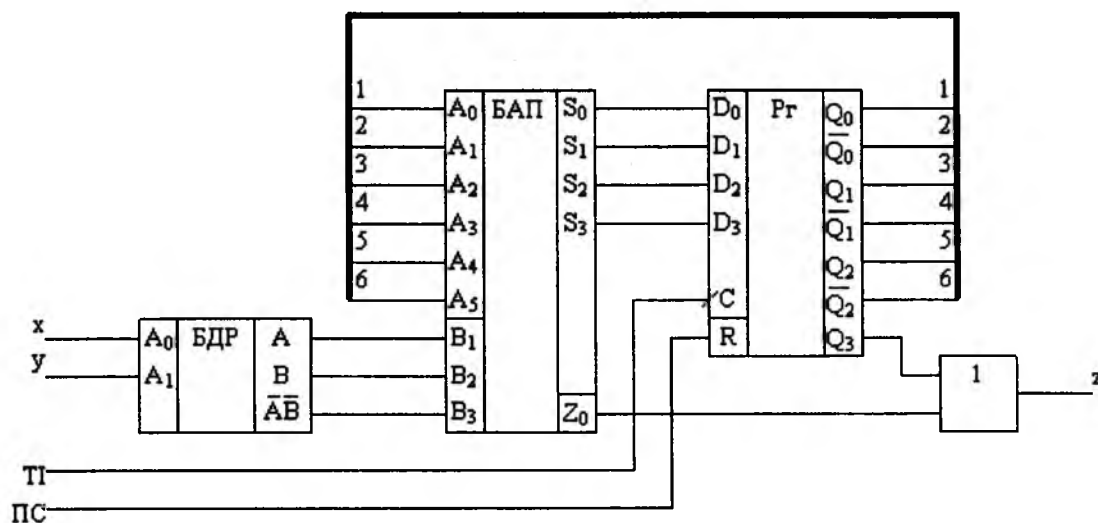


Рисунок 5 – Функціональна схема порозрядного суматора послідовних кодів золотої пропорції

Для розробки першого варіанту функціональної схеми БАП скористаємося схемою визначення допоміжних сигналів А і В, поданою на рис 4. Шляхом аналізу переходів між станами можна визначити такі логічні вирази для D-входів тригерів регістра та допоміжного сигналу виходу:

$$\begin{aligned} D1 &= \overline{Q3} \overline{Q1} A + Q2 B; \\ D2 &= Q1 \overline{A} \overline{B}; \\ D3 &= Q2 \overline{A} \overline{B} + \overline{Q1} \overline{Q2} \overline{Q3} B + \overline{D1} (Q3 + Q4) A; \\ D4 &= Q3 \overline{A} \overline{B} + D1 \overline{Q4} \overline{Q2} + (Q1 + Q2) \overline{Q4} B; \\ D5 &= Q4 \overline{A} \overline{B} + (Q3 + Q4) \overline{Q1} B + \overline{Q4} D4 A; \\ Z0 &= Q4 Q1 B + Q4 A. \end{aligned}$$

Перший варіант функціональної схеми блоку універсального АЛ-перетворення подано на рисунку 6. Дана схема блоку адитивного перетворення реалізована на 21 логічному елементі.

У другому варіанті зменшення розрядності регістра досягається перекодуванням його станів. Варіантів може бути декілька. Оберемо такий варіант перекодування:

$$\begin{aligned} 0000 &\rightarrow 000; \\ 0001 &\rightarrow 001; \\ 0010 &\rightarrow 010; \\ 0100 &\rightarrow 011; \\ 1000 &\rightarrow 100; \\ 1001 &\rightarrow 101. \end{aligned}$$

Схема, що реалізує дану таблицю, фактично є схемою UAL- перетворення для кодів золотої пропорції. Виконуючи мінімізацію диз'юнктивних форм, можна визначити такі логічні вирази для D-входів тригерів регістра:

$$\begin{aligned} D1 &= \overline{Q1} \overline{Q2} \overline{Q3} A + \overline{Q1} B + \overline{Q1} Q2 \overline{A} \overline{B}; \\ D2 &= Q1 \overline{Q2} \overline{Q3} A + \overline{Q1} \overline{Q2} B + \overline{Q1} Q2 \overline{Q2} \overline{Q3} B + \overline{Q1} \overline{Q2} \overline{A} \overline{B}; \\ D3 &= \overline{Q1} \overline{Q2} \overline{Q3} A + \overline{Q1} Q2 \overline{Q3} B + Q1 \overline{Q2} \overline{Q3} B + Q1 Q2 \overline{A} \overline{B}; \\ D4 &= \overline{Q3} \overline{Q1} \overline{Q2} A + Q1 Q2 B + \overline{Q1} \overline{Q2} Q3 B + Q3 \overline{A} \overline{B}; \\ Z0 &= Q3 A + Q1 \overline{Q2} Q3 B. \end{aligned}$$

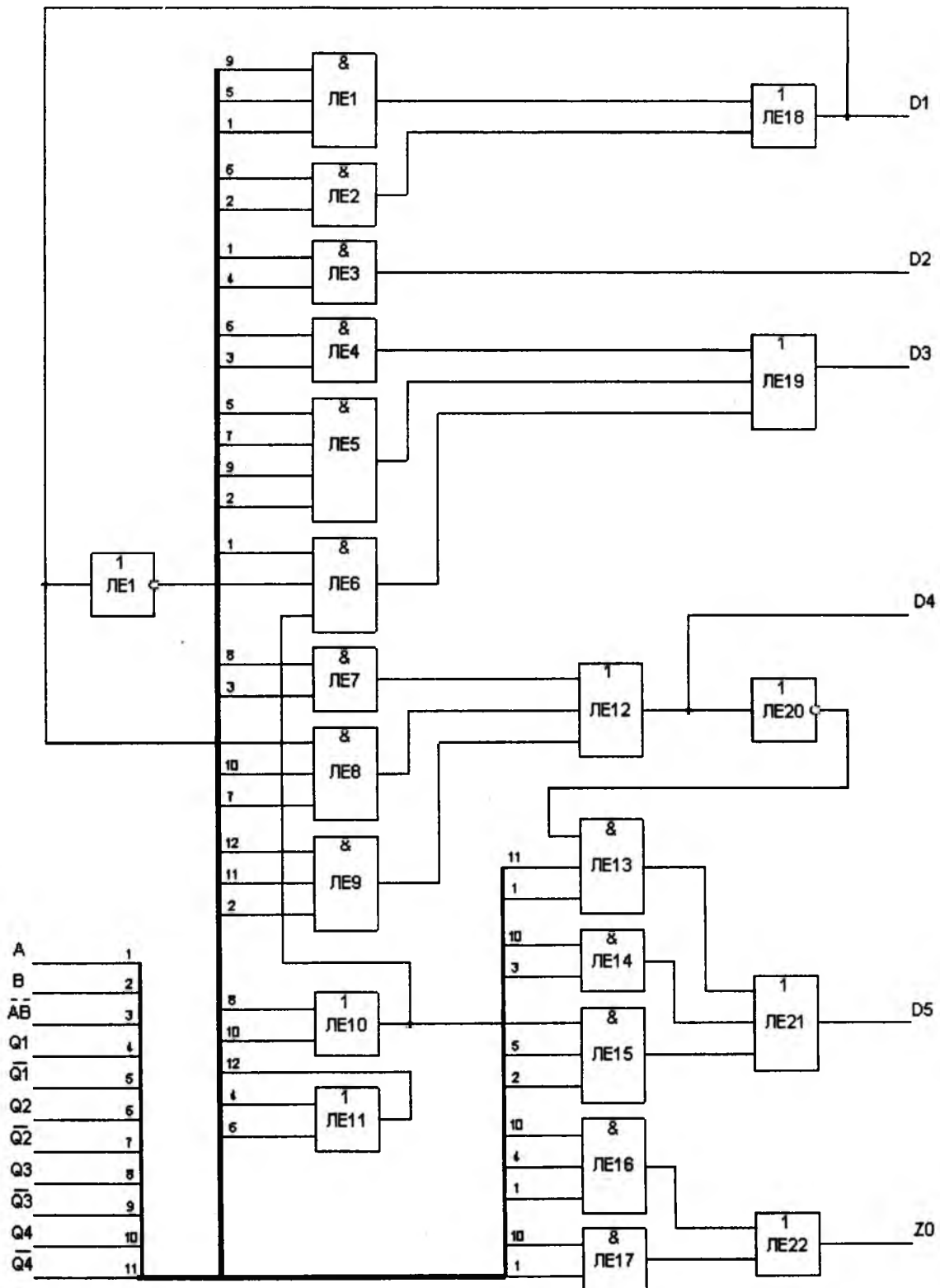


Рисунок 6 – Функціональна схема блоку універсального адитивного перетворення зі зменшеними апаратними витратами

Для зменшення апаратних витрат на реалізацію логічних виразів для D-входів тригерів введемо допоміжні сигнали:

$$\begin{aligned}
 S1 &= \overline{Q1} B; \\
 S2 &= S1 \overline{Q2}; \\
 S3 &= S2 Q3;
 \end{aligned}$$

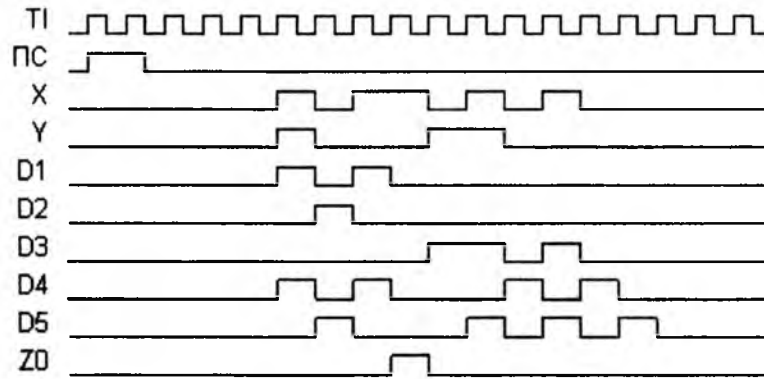


Рисунок 8 – Часові діаграми роботи суматора у першому варіанті

У другому варіанті процес порозрядного додавання даних кодів являє собою таку послідовність операцій:



На рисунку 9 подано часові діаграми роботи суматора у другому варіанті, отримані шляхом програмного моделювання.

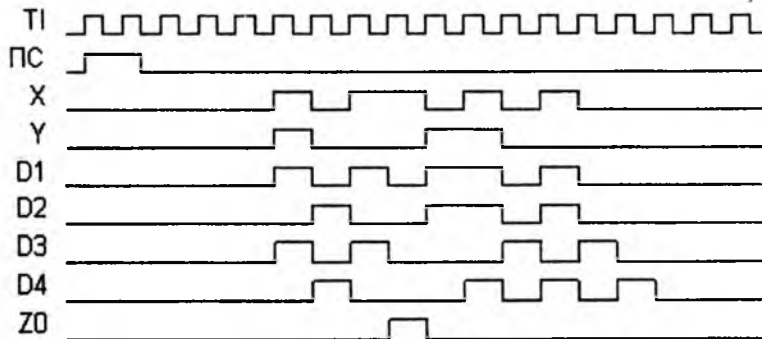


Рисунок 10 – Часові діаграми роботи суматора у другому варіанті

Дані часові діаграми співпадають з результатами аналізу виконання прикладів для першого і другого варіантів. Отже, часові діаграми підтверджують правильність роботи обох варіантів побудови суматора.

Висновки

В даній статті описані результати, отримані у процесі розробки алгоритмів побітового виконання додавання, віднімання, множення та ділення послідовних кодів золотої пропорції.

1. Вперше розроблені структурна та функціональна схеми побітового додавання кодів золотої пропорції на основі елементарних адитивних перетворень. При цьому досягається найменша розрядність регістра проміжного результату.

2. Вперше розроблені два варіанти реалізації функціональної схеми блоку адитивного перетворення: перший – зі зменшеною розрядністю, другий – зі зменшеними апаратними витратами. Кожен з варіантів є ефективним, але у різних випадках.

3. Вперше проведено програмне моделювання варіантів побудови суматора, що підтвердило правильність розроблених схем.

Список літератури

1. Avizenis A. Binary-compatible signet-digit arithmetic. IN: AFIPS Conf Proc. – Vol. 26 – P1. – 1964 – P.663.
2. Самофалов К.Г., Луцкий Г.М. Основы построения конвейерных ЭВМ.- Киев: Вища школа, 1981. – 234 с.
3. Каляев А.В. Многопроцессорные системы с программируемой архитектурой. – М.: Радио и связь, 1984. – 240 с.
4. Ch. Frougny, On-line finite automata for addition in some numeration systems. Theoretical Informatics and Applications 33 (1999), 79–101.
5. Стахов А.П. Коды золотой пропорции. – М.: Радио и связь, 1983. – 152с., ил. – (Кибернетика).
6. Системи числення з адитивними та мультиплікативними співвідношеннями між вагами розрядів / О.Д.Азаров, О.І.Черняк, П.О.Черняк // Вісник ВПІ. – 2001. - №1. – С. 58-64.
7. Визначення довжини перенесення при додаванні у системах числення з адитивними та мультиплікативними співвідношеннями між вагами розрядів / О.Д. Азаров, О.І. Черняк // Наукові праці Донецького національного університету. Серія: "Обчислювальна техніка та автоматика". Випуск 74 – Донецьк: ДонНТН, 2004, С 401-408.
8. Методи конвеєрної порозрядної обробки послідовних кодів золоті пропорції / О.І. Черняк, О.Д. Азаров // Вісник ВПІ. – 1996. - №1. – С. 14-17.
9. Алгоритмічні основи побітової обробки кодів золоті пропорції / О.Д. Азаров, О.І. Черняк //

Азаров Олексій Дмитрович, д.т.н., проф., директор інституту інформаційних технологій та комп'ютерної інженерії, зав. кафедри обчислювальної техніки Вінницького національного технічного університету, 21021, м. Вінниця, вул. Порики, 16, кв. 13. т.(0432)43-90-02 дом. 43-75-07, azarov@lili.vstu.vinnica.ua.

Черняк Олександр Іванович, ст. викладач кафедри обчислювальної техніки Вінницького національного технічного університету, 21021, м. Вінниця, вул. Келецька, 94, кв. 40, т.(0432)43-90-02, дом. 46-37-16, chernyak@vstu.vinnica.ua.