

ТЕОРЕТИЧНІ ЗАСАДИ ТА ПРИНЦИПИ ПОБУДОВИ АРИФМЕТИКО-ЛОГІЧНОГО ПРИСТРОЮ НА ОСНОВІ ВЕРТИКАЛЬНО-ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

Описані теоретичні засади, структура та принципи роботи арифметико-логічного пристрою. Розглянемо функції і алгоритми виконання математичних операцій. Досліджено приклади виконання арифметичних операцій в базисах Радемахера і Галуа. Виконана порівняльна характеристика суматорів в базисі Радемахера та Галуа.

We are describe the theoretical background, structure and principles of arithmetic logic unit. Research function and algorithm of mathematical operations. Arithmetic operations research as examples in Rademacher's basis and Galois' basis. The comparative characteristics of adders in Rademacher's basis and Galois' basis.

Ключові слова: арифметико-логічний пристрій, вертикально-інформаційна технологія, теоретико-числовий базис.

Вступ

У зв'язку з інтенсивним впровадженням обчислювальної техніки в усі сфери сучасного суспільства одним із пріоритетних завдань економічного та соціального розвитку є автоматизоване проектування нових мікропроцесорних систем, у тому числі арифметико-логічних пристроїв (АЛП). Сучасний розвиток мікропроцесорної індустрії вимагає нових теоретичних, та практичних науково-технічних рішень.

Постановка проблеми. Методологічною основою при розробці більшості процесорів являється двійкова система числення в теоретико-числовому базисі (ТЧБ) Радемахера. Дослідження тенденції розвитку процесорів обумовленні теоретичним вичерпанням застосування базису Радемахера для побудови арифметико-логічних пристроїв процесора, до яких ставляться все жорсткіші вимоги щодо структурної побудови, енергоспоживання та вартості. Світовий досвід за останні роки демонструє тенденції у дослідженні інших базисів: Галуа, Радемахера, Крестенсона, та унітарного, які також породжують системи числення [1]. У зв'язку з цим досить актуальним для побудови АЛП процесора являється базис Галуа, який є основою вертикально-інформаційної технології (ВІТ) [2]. Порівняння функціональних можливостей АЛП у базисах Радемахера та Галуа дозволить знайти ефективні рішення для розробки процесорних пристроїв з вдосконаленими характеристиками.

Аналіз останніх досліджень і публікацій. Значний внесок в розвиток теорії та застосування в галузі обчислювальних машин в різних теоретико-числових базисах зробили Акушський І.Й., Крестенсон А., Глушков В.М., Палагін О.В., Брюхович Е.И., Романов С.І., Тарасенко В.П., Николайчук Я.М., Мельник А.О., Петришин Л.Б., Червяков В.П. та інші [3–9]

Постановка завдання. Основною задачею наукового дослідження є вирішення наступних науково-технічних завдань:

1. Дослідження функцій АЛП процесора.
2. Алгоритм виконання арифметичних операцій в базисі Радемахера та Галуа.
3. Дослідження кодів Галуа, приклади виконання арифметичних операцій.
4. Дослідження суматорів на основі ТЧБ Радемахера і Галуа.

Мета роботи – розробити теоретичні засади та принципи побудови арифметико-логічного пристрою у теоретико-числовому базисі Галуа; зробити порівняльну характеристику виконання операцій в суматорах на основі базисів Радемахера та Галуа.

Функції і алгоритм виконання операцій в АЛП. У цифровій апаратурі основним пристроєм, у якому безпосередньо виконується обробка, є процесорний пристрій. Процесор складається з двох основних компонентів: пристрій керування та АЛП. Арифметико-логічний пристрій призначений для виконання арифметичних і логічних операцій над числами (словами), що надходять до нього, за сигналами з пристрою керування. Пристрій керування призначений для організації процесу обчислень. Він координує дії АЛП, генеруючи у визначеній часовій послідовності керуючі сигнали, під дією яких у вузлах АЛП виконуються необхідні операції.

Розглянемо спрощену блок-схему роботи АЛП (рис.1) Всі дані в арифметико-логічному модулі і накопичувальний регістр (акумулятор) надсилаються через регістр даних. Накопичувальний регістр має розмір, що відповідає довжині машинного слова. Для того, щоб скласти два двійкових числа, одне число запам'ятовується у накопичувальному регістрі, а інше - запам'ятовується у регістрі даних. Після додавання сума двох чисел одноадресної мікрокоманди:

$$A := A * R \text{ або } R_2 := A * R_1, \quad (1)$$

де * – символ арифметико-логічної операції (+, -, x, /, <, >, =, ≤, ≥, ⊕, ↔).

Процес функціонування АЛП розпадається на певну послідовність елементарних дій у його вузлах. Перелік таких елементарних дій містить у собі:

- 1) встановлення регістра в деякий стан;

- 2) інвертування вмісту розрядів регістру;
- 3) пересилку вмісту одного вузла в інший вузол (наприклад, пересилку числа з регістра в регістр);
- 4) зсув вмісту вузла (регістра) ліворуч або праворуч;
- 5) виконання арифметико-логічних операцій.

Реалізуючи операції двійкової арифметики структура АЛП визначається алгоритмами виконання операцій. На вибір алгоритмів найбільше впливають вимоги до швидкодії АЛП чи обмеження на кількість використовуваного обладнання.

Операція додавання двійкових чисел є основною операцією, вона найчастіше використовується в програмах і багаторазово використовується в операціях множення і ділення. Операції додавання в АЛП можуть виконуватись над двійковими числами з фіксованою і плаваючою комами. В окремих випадках операції можуть виконуватись над кодами цілих чисел. В основному використовуються два основних способи реалізації додавання [7]: 1) зворотному коді; 2) доповнючому коді. Для контролю за переповненням розрядної сітки акумулятора використовують модифіковані коди, в яких знак представляється двома двійковими розрядами: знак плюс – кодом 00 і знак мінус – кодом 11. Зворотній або доповнюючий код використовують для кодування від’ємних операндів.

Порівняльна характеристика суматорів в базисі Радемахера та Галуа

Головним недоліком виконання операції над двійковими кодами в базисі Радемахера є наявність наскрізних порозрядних переносів. В базисі Галуа, як показано в роботах [10] реалізується рекурентний спосіб представлення чисел згідно з прикладом представленим у табл. 1 коду поля Галуа $G \begin{pmatrix} 4 \\ 2 \end{pmatrix}$

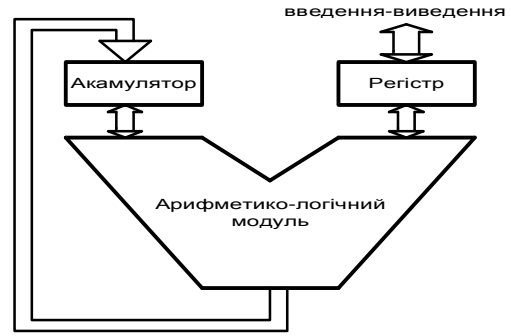


Рис. 1. Структурна схема роботи АЛП

Таблиця 1

Формування кодів чисел в базисах Радемахера та Галуа

| Десяткове значення | Двійкові коди | Коди Галуа | Вектор формування кодів Галуа |
|--------------------|---------------|------------|-------------------------------|
| 0 | 0000 | 1111 | 1 |
| 1 | 0001 | 1110 | 1 |
| 2 | <0010> → | <1101> → | → 1 |
| 3 | 0011 | 1010 | 1 |
| 4 | 0100 | 0101 | 0 |
| 5 | 0101 | 1011 | 1 |
| 6 | 0110 | 0110 | 0 |
| 7 | 0111 | 1100 | 1 |
| 8 | 1000 | 1001 | 1 |
| 9 | 1001 | 0010 | 0 |
| 10 | 1010 | 0100 | 0 |
| 11 | 1011 | 1000 | 1 |
| 12 | 1100 | 0000 | 0 |
| 13 | 1101 | 0001 | 0 |
| 14 | 1110 | 0011 | 0 |
| 15 | 1111 | 0111 | 0 |

Таким чином згідно з табл.1, матриця кодів чисел в базисі Радемахера відповідає упакованим кодам в базисі Галуа. Базис Галуа формується на основі вектора кодів Галуа, як показано у табл.1 десятковому значенню 2 відповідає код Галуа 1101, на рис.2 шляхом циклічного зсуву формується значення 0011 коду Галуа, яке відповідає двійковому значенню 1110, і десятковому значенню 14.

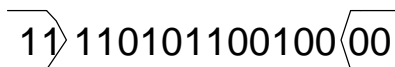


Рис. 2. Принцип формування кодонів Галуа на границі бітів

Причому вектор кодів Галуа для подання всієї послідовності чисел 2^k повинен замикатися в кільце, на границі стикування якого формуються відповідні коди. Для генерації рекурентних кодів чисел базису Галуа використовуються кодові ключі, які визначаються на основі незвідних поліномів [11]. Наприклад, для

приведеного коду Галуа $G \begin{pmatrix} 4 \\ 2 \end{pmatrix}$ алгоритм генерування кодів Галуа описується рівнянням:

$$G_{i+1} = G_i \oplus G_{i-3}, \tag{3}$$

де G_i – біти Галуа,

\oplus – операція по модулю 2.

Виходячи з рівняння (3) кожен $i+1$ біт Галуа формується на основі логічної операції за модулем 2 над попередніми бітами Галуа. Таким чином операція інкремента в базисі Галуа виконується у виді зсуву вліво кодону, який представляє число в базисі Галуа. Наприклад, x_i у базисі Радемахера та Галуа представлено кодами згідно з табл. 1.

$$X_i = 11_{(10)} = 1011_{(R)}; X_i = 11_{(10)} = 1000_{(G)}.$$

В базисі Радемахера відомий ряд структур суматорів [10], які показані на (рис. 3). Робота даних суматорів описується рівняннями:

- для півсуматора $S = A \wedge \bar{B} \vee \bar{A} \wedge B$; $P = A \wedge B$;

- для повного суматора

$$S_i = \bar{A}_i \wedge \bar{B}_i \wedge P_{i-1} \vee \bar{A}_i \wedge B_i \wedge \bar{P}_{i-1} \vee A_i \wedge \bar{B}_i \wedge P_{i-1} \vee A_i \wedge B_i \wedge \bar{P}_{i-1},$$

$$P_i = \bar{A}_i \wedge B_i \wedge P_{i-1} \vee A_i \wedge \bar{B}_i \wedge P_{i-1} \vee A_i \wedge B_i \wedge \bar{P}_{i-1} \vee A_i \wedge B_i \wedge P_{i-1}.$$

Нарощування розрядності виконується послідовним з'єднанням однорозрядних повних суматорів (рис. 3 в) залежно від необхідної розрядності. Основним недоліком суматорів в базисі Радемахера є наявність наскрізного переносу, що призводить до значного зниження швидкодії процесорів, яке пропорційне їх розрядності і часу затримки сигналів згідно виразу:

$$T_{\Sigma R} = kT_c, \quad T_c = 3T_{ле} \tag{3}$$

де kT_c – розрядність суматора,

T_c – час спрацювання повного суматора,

$T_{ле}$ – час спрацювання логічних елементів.

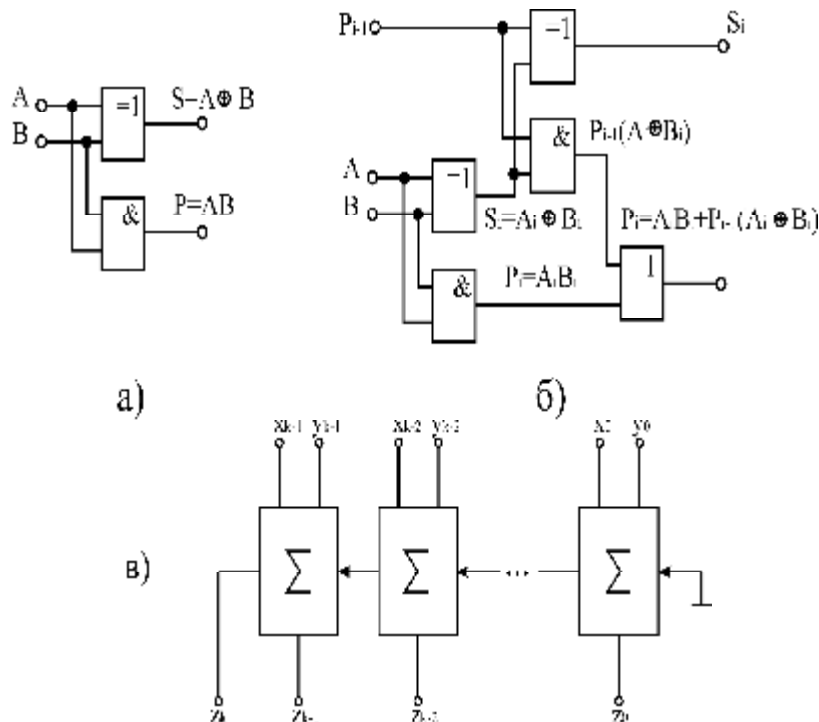


Рис. 3. Структури двійкових однорозрядних суматорів (а – півсуматор, б – повний суматор, в – k-розрядний суматор)

Граничним випадком максимального зниження швидкодії таких суматорів є операція інкременту чи декременту згідно арифметичної операції у базисі Радемахера:

$$\begin{array}{r} \leftarrow\leftarrow\leftarrow\leftarrow\leftarrow\leftarrow \\ 111\dots11 \quad X_{(2)}=2^{k-1}; \\ + 000\dots01 \quad Y_{(2)}=2^0; \\ \hline 1000\dots00 \quad Z_{(2)}=2^k, \end{array}$$

Рис. 4. Принцип виконання операції інкременту, де \leftarrow – символ наскрізного переносу між розрядами суматора

Слід відмітити, що дана операція інкрементного сумування в базисі Галуа виконується за один такт незалежно від розрядності суматора на базі структури представленої на рис. 4, де логічні елементи виконують функції рекурентного формування $i+1$ -х бітів коду Галуа. При цьому час затримки сигналів при виконанні операції інкременту сумування визначається згідно виразу:

$$T_{\Sigma GI} = T_T + 3kT_{le} \quad (4)$$

Перевагою інкрементного суматора в базисі Галуа є наявність пам'яті, в той час, як суматор в базисі Радемахера для зберігання даних потребує 3-х k -розрядних регістрів.

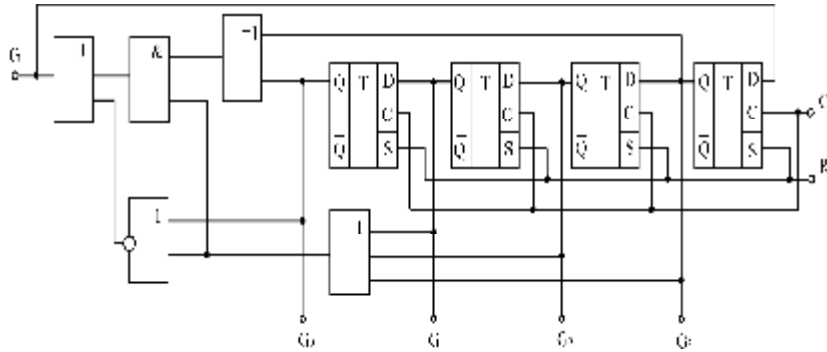


Рис. 5. Структура 4-х розрядного інкрементного суматора в базисі Галуа

Операція сумування кодів у базисі Галуа, для суматорів в діапазоні 2^k-1 , виконується шляхом виконання логічних операцій над бітами коду Галуа першого з доданків згідно таблиці логічних рівнянь, що описують другий доданок. В табл. 2 надані формули логічного опису бітів другого доданку, на прикладі 4-розрядного коду Галуа для суматорів в діапазоні 2^k , що досягається введенням додаткової комбінації "0000" в позиції 12 [11].

Таблиця 2

Формули логічного представлення кодів Галуа

| Десятькове значення | Код Галуа | Формула суматора | | | |
|---------------------|-----------|--|--|--|--|
| | | b_4 | b_3 | b_2 | b_1 |
| 0 | 1111 | b_4 | b_3 | b_2 | b_1 |
| 1 | 1110 | b_3 | b_2 | b_1 | $b_1 \oplus b_4 \oplus b_3$ |
| 2 | 1101 | b_3 | b_1 | b_2 | $b_1 \oplus b_4$ |
| 3 | 1010 | b_2 | b_2 | $b_1 \oplus b_4$ | $b_1 \oplus b_2 \oplus b_4$ |
| 4 | 0101 | b_1 | $b_1 \oplus b_4$ | $b_1 \oplus b_2 \oplus b_4$ | $b_1 \oplus b_2 \oplus b_3 \oplus b_4$ |
| 5 | 1011 | $b_1 \oplus b_4$ | $b_1 \oplus b_2 \oplus b_4$ | $b_1 \oplus b_2 \oplus b_3 \oplus b_4$ | $b_1 \oplus b_2 \oplus b_3$ |
| 6 | 0110 | $b_1 \oplus b_2 \oplus b_4$ | $b_1 \oplus b_2 \oplus b_3 \oplus b_4$ | $b_1 \oplus b_2 \oplus b_3$ | $b_2 \oplus b_3 \oplus b_4$ |
| 7 | 1100 | $b_1 \oplus b_2 \oplus b_3 \oplus b_4$ | $b_1 \oplus b_2 \oplus b_3$ | $b_2 \oplus b_3 \oplus b_4$ | $b_1 \oplus b_3$ |
| 8 | 1001 | $b_1 \oplus b_2 \oplus b_3$ | $b_2 \oplus b_3 \oplus b_4$ | $b_1 \oplus b_3$ | $b_2 \oplus b_4$ |
| 9 | 0010 | $b_2 \oplus b_3 \oplus b_4$ | $b_1 \oplus b_3$ | $b_2 \oplus b_4$ | $b_1 \oplus b_3 \oplus b_4$ |
| 10 | 0100 | $b_1 \oplus b_3$ | $b_2 \oplus b_4$ | $b_1 \oplus b_3 \oplus b_4$ | $b_1 \oplus b_2$ |
| 11 | 1000 | $b_2 \oplus b_4$ | $b_1 \oplus b_3 \oplus b_4$ | $b_1 \oplus b_2$ | $b_2 \oplus b_3$ |
| 12 | 0000 | $b_1 \oplus b_3 \oplus b_4$ | $b_1 \oplus b_2$ | $b_2 \oplus b_3$ | $b_3 \oplus b_4$ |
| 13 | 0001 | $b_1 \oplus b_2$ | $b_2 \oplus b_3$ | $b_3 \oplus b_4$ | b_4 |
| 14 | 0011 | $b_2 \oplus b_3$ | $b_3 \oplus b_4$ | b_4 | b_3 |
| 15 | 0111 | $b_3 \oplus b_4$ | b_4 | b_3 | b_2 |

Такий спосіб опису функцій суматора в базисі Галуа передбачає емуляцію його роботи виключно програмним шляхом, що не дозволяє перейти його апаратної реалізації. Виконання операції сумування на основі матриці коефіцієнтів d_{ij} (табл. 3) представлено в [1], яка використовується для логічного формування бітів коду Галуа суми доданків згідно виразу:

$$b_i = d_{i,k} \cdot b_k \oplus d_{i,k-1} \cdot b_{k-1} \oplus \dots \oplus d_{i,1} \cdot b_1 \quad (5)$$

Матриця коефіцієнтів d_{ij}

| Десяткове значення | Код Галуа | Формула суматора | | | |
|--------------------|-----------|------------------|----------|----------|----------|
| | | d_{j4} | d_{j3} | d_{j2} | d_{j1} |
| 0 | 1111 | 1000 | 0100 | 0010 | 0001 |
| 1 | 1110 | 0100 | 0010 | 0001 | 1001 |
| 2 | 1101 | 0010 | 0001 | 1001 | 1011 |
| 3 | 1010 | 0001 | 1001 | 1011 | 1111 |
| 4 | 0101 | 1001 | 1011 | 1111 | 0111 |
| 5 | 1011 | 1011 | 1111 | 0111 | 1110 |
| 6 | 0110 | 1111 | 0111 | 1110 | 0101 |
| 7 | 1100 | 0111 | 1110 | 0101 | 1010 |
| 8 | 1001 | 1110 | 0101 | 1010 | 1101 |
| 9 | 0010 | 0101 | 1010 | 1101 | 0011 |
| 10 | 0100 | 1010 | 1101 | 0011 | 0110 |
| 11 | 1000 | 1101 | 0011 | 0110 | 1100 |
| 12 | 0000 | 0011 | 0110 | 1100 | 1000 |
| 13 | 0001 | 0110 | 1100 | 1000 | 0100 |
| 14 | 0011 | 1100 | 1000 | 0100 | 0010 |
| 15 | 0111 | 0111 | 1110 | 1101 | 0011 |

Розглянемо приклад виконання операції додавання двох чисел в базисі Галуа на основі матриці коефіцієнтів d_{ij} . Нехай $X_{(10)}=2$; $Y_{(10)}=5$, тоді $X_G=1101$; $Y_G=1011$. Тобто X_G відповідає коду $b_4=1$; $b_3=1$; $b_2=0$; $b_1=1$, а код Y_G згідно табл. 3 відповідає логічним операціям над бітами X_G : $b_1 \oplus b_2 \oplus b_4$; $b_1 \oplus b_2 \oplus b_3 \oplus b_4$; $b_1 \oplus b_2 \oplus b_3$; $b_2 \oplus b_3 \oplus b_4$, що відповідає кодам d_{ij} 1011; 1111; 0111; 1110 з табл. 2. Тобто результат сумування даних чисел виконується за допомогою логічної обробки кодів X_G та коефіцієнтів d_{ij} , які відповідають коду Y_G :

$$G_4 = 1 \wedge b_4 \oplus 0 \wedge b_3 \oplus 1 \wedge b_2 \oplus 1 \wedge b_1 = 1 \wedge 1 \oplus 0 \wedge 1 \oplus 1 \wedge 0 \oplus 1 \wedge 1 = 1;$$

$$G_3 = 1 \wedge b_4 \oplus 1 \wedge b_3 \oplus 1 \wedge b_2 \oplus 1 \wedge b_1 = 1 \wedge 1 \oplus 1 \wedge 1 \oplus 1 \wedge 0 \oplus 1 \wedge 1 = 1;$$

$$G_2 = 0 \wedge b_4 \oplus 1 \wedge b_3 \oplus 1 \wedge b_2 \oplus 1 \wedge b_1 = 0 \wedge 1 \oplus 1 \wedge 1 \oplus 1 \wedge 0 \oplus 1 \wedge 1 = 0;$$

$$G_1 = 1 \wedge b_4 \oplus 1 \wedge b_3 \oplus 1 \wedge b_2 \oplus 0 \wedge b_1 = 1 \wedge 1 \oplus 1 \wedge 1 \oplus 1 \wedge 0 \oplus 0 \wedge 1 = 0.$$

Отримана система логічних рівнянь дозволяє синтезувати структуру 4-бітового суматора Галуа, який зображений на рис. 6.

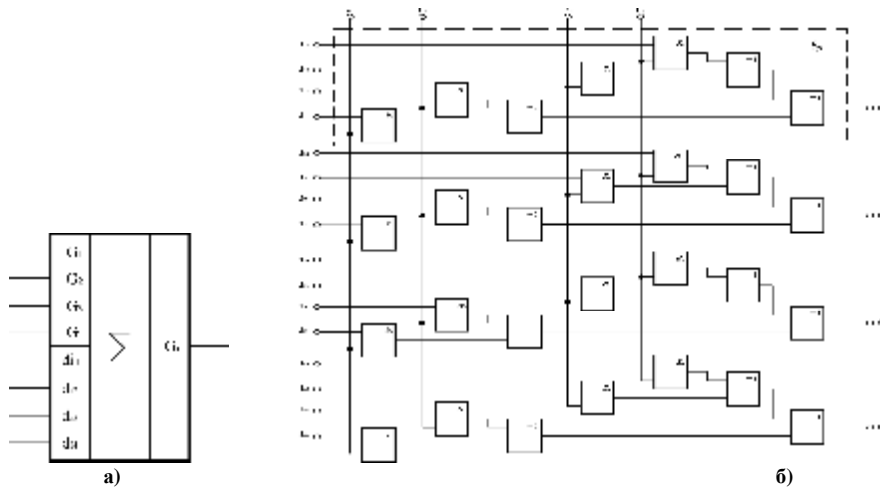


Рис. 6. Умовне позначення (а) одного розряду 4-бітового суматора Галуа та структурна схема 4-розрядного суматора Галуа (б)

Аналіз структури операційного пристрою сумування в базисі Галуа показує, що він характеризується регулярною архітектурою з часом затримки сигналів:

$$T_{\Sigma G} = 3T_{ле} + \left(\frac{k}{4}\right)T_{ле}. \tag{6}$$

Для функціонування даного операційного пристрою необхідно виконувати дешифрування коду k -розрядного коду другого доданку в k^2 -розрядний код коефіцієнтів d_{ij} . На рис. 7 зображено умовне позначення (а) та розроблена канонічна структура дешифратора суматора Галуа (б), яка характеризується часом затримки згідно з виразом:

$$T_{D\Sigma} = 3T_{ле} \tag{7}$$

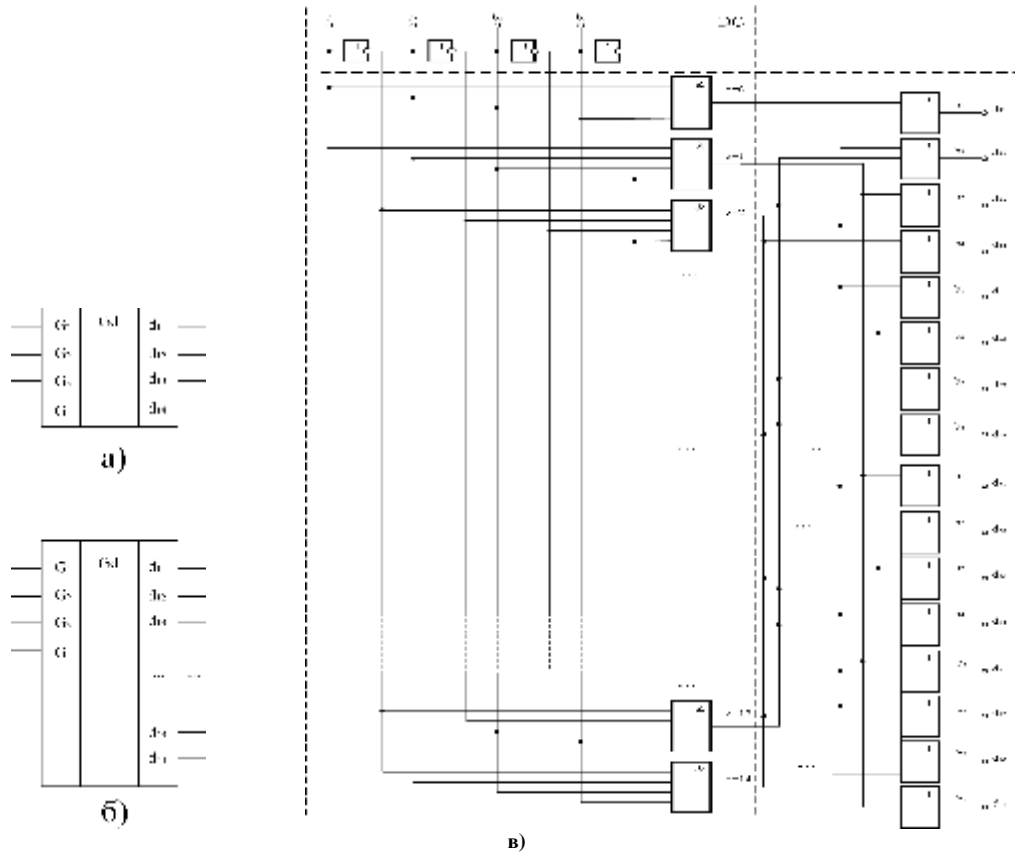


Рис. 7. Умовне позначення однорозрядного (а), 4-розрядного (б) та структура дешифратора 4-розрядного суматора Галуа (в)

При використанні умовних позначень компонентів суматора Галуа (рис. 6 а, 7 б), отримаємо структурну схему 4-розрядного паралельного суматора Галуа, яка показана на рис. 8.

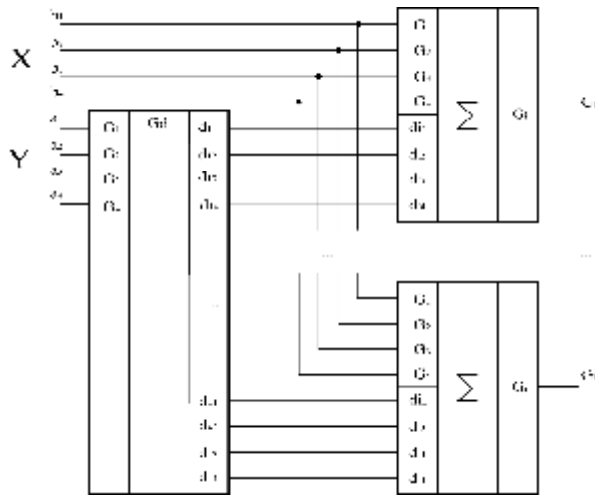


Рис. 8. Структурна схема 4-розрядного паралельного суматора Галуа

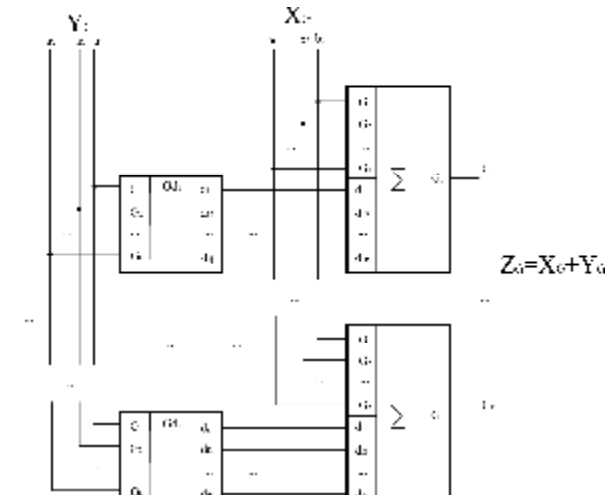


Рис. 9. Структурна схема k розрядного паралельного суматора Галуа

При використанні k-розрядних фрагментів дешифраторів (рис. 7 а) та однорозрядних суматорів (рис. 6 а) структура паралельного суматора k-розрядного паралельного суматора Галуа отримає вид (рис. 9).

Згідно виразів (6), (7) час затримки сигналів при виконання операції сумування в базисі Галуа на основі структури рис. 7 дорівнює:

$$T_{\Sigma G} = 5T_{ле} + kQ_{ле}, \tag{8}$$

де $Q_{ле} = \frac{T_{ле}}{4}$, оскільки дана затримка відповідає 4-м розрядам суматора, а при подвоєнні розрядності суматора в схему послідовно включається один додатковий елемент "виключаюче АБО", затримка сигналів

$5T_{ле}$, яка включає 3 послідовні елементи однорозрядного суматора Галуа та 2 елементи дешифратора.

Оцінка швидкодії суматорів в базисі Радемахера і Галуа розраховується за формулами:

$$V_{\Sigma R} = \frac{1}{3kT_{ле}}, V_{\Sigma G} = \frac{1}{5T_{ле} + kQ_{ле}}. \quad (9)$$

Звідки відносна оцінка швидкодії суматора в базисі Галуа в порівнянні з реалізацією суматора в базисі Радемахера розраховується згідно з виразом:

$$V = \frac{3kT_{ле}}{5T_{ле} + kQ_{ле}}. \quad (10)$$

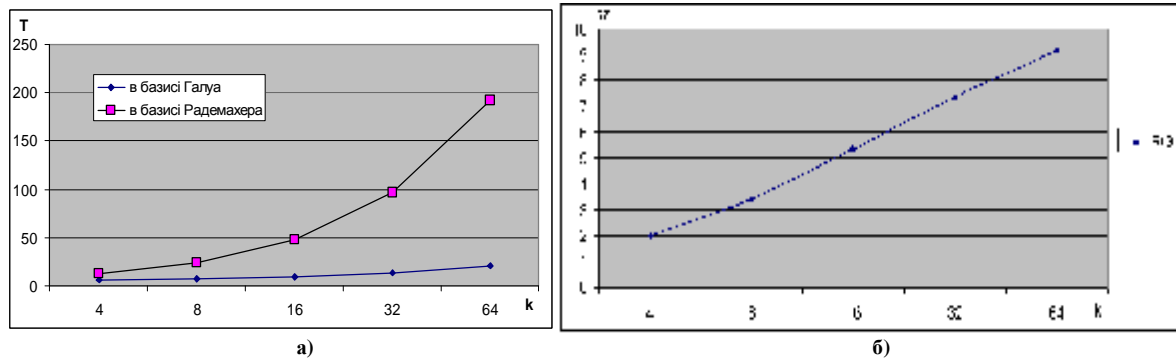


Рис. 10. Характеристики часу затримки паралельних суматорів в базисах Радемахера та Галуа (а) та відносна оцінка швидкодії суматора Галуа залежно від розрядності процесора (б)

Оскільки діапазон кодування чисел в суматорах Галуа не перевищує 2^k обов'язковою вимогою правильної роботи даного класу суматорів без переповнення розрядної сітки є умова $X_G + Y_G < 2^k$. З графіка (рис. 8, б) можна побачити, що при побудові процесорів з розрядністю $k=16-64$ швидкодія суматорів в базисі Галуа перевищує швидкодію суматорів в базисі Радемахера в 5–9 разів.

Висновок. Викладені теоретичні засади виконання операції сумування в базисах Радемахера і Галуа показують, що в базисі Галуа внаслідок відсутності міжрозрядних переносів зростає швидкодія процесорів в залежності від їх розрядності. Крім того суматори в базисі Галуа характеризуються регулярністю структури, що спрощує їх мікроелектронну реалізацію на базі програмованих логічних матриць. Звідси постає подальша перспектива проектування спецпроцесорів і його компонентів на основі вертикально-інформаційної технології.

Література

1. Николайчук Я.М. Теория джерел інформації / Николайчук Я.М. – Тернопіль : ТзОВ Терно-Граф. – 2010–534 с.
2. Николайчук Я.М. Дослідження архітектури комп'ютерів: принципи побудови процесорів на основі вертикально-інформаційної технології Поступ в науку / Я.М. Николайчук, П.В. Гуменний // Збірник наукових праць Бучацького інституту менеджменту і аудиту. – Бучач. – 2009. – № 5. – С. 69–73.
3. Палагин А.В. Реконфигурируемые структуры на ПЛИС / А.В. Палагин, В.Н. Опанасенко, В.Г. Сахарин // УсиМ. – 2000. – № 3. – С. 33–43.
4. Петришин Л.Б. Теоретичні основи перетворення форми та цифрової обробки інформації в базисі Галуа : [навч. посібник] / Петришин Л.Б. – К. : ІЗІМН МОУ, 1997. – 237 с.
5. Глушков В.М. Основы безбумажной информатики / Глушков В.М. – М. : Наука, 1987. – 552 с.
6. Акушский И.Я. Машинная арифметика в остаточных классах / И.Я. Акушский, Д.И. Юдицкий. – М. : Советское радио, 1968. – 440 с.
7. Тарасенко В.П., Основы компьютерной арифметики / В.П. Тарасенко В.И. Корнейчук. – К. : Вища школа, 2003. – С. 34–56.
8. Модулярные параллельные вычислительные структуры нейропроцессорных систем / Н.И. Червяков, С.А. Рядное, П.А. Сахнюк, А.В. Шапошников. – М. : ФИЗМАТ ЛИТ, 2003. – 288 с.
9. Мельник А.О. Архітектура комп'ютера / Мельник А.О. – Луцьк : Видавництво обласної друкарні 2008 р. – 468 с.
10. Угрюмов Е.П. Цифровая схмотехника / Угрюмов Е.П. – СПб. : БХВ – Санкт-Петербург, 2002. – 528 с.
11. Николайчук Я.М. Дослідження системних характеристик двомірних кодів з особливими кореляційними властивостями / Я.М. Николайчук, О.М. Заставний // Вісник технологічного університету Поділля, Хмельницький, 2004. – № 2. – Ч. 1. – Т 2. – С. 107–110.

Рецензент: д.т.н. Горбійчук М.І.
Надійшла 7.2.2012 р.