

УДК 681.325

Замрій Б.А., старший викладач кафедри обчислювальної техніки,  
НУ водного господарства та природокористування

## ДОСЛІДЖЕННЯ ДІАПАЗОНУ ЧИСЕЛ АЦИКЛІЧНОЇ МОДЕЛІ СУМАТОРА З ЛОГІЧНИМИ ЕЛЕМЕНТАМИ OR ТА XOR В ОСТАННЬОМУ РОЗРЯДІ

**Замрій Б.А.** Дослідження діапазону чисел ациклічної моделі суматора з логічними елементами OR та XOR в останньому розряді. Проведеними дослідженнями встановлена перспектива збільшення продуктивності обчислення сигналів суми і перенесення ациклічних суматорів бінарних кодів з логічними елементами OR в останньому розряді. Зв'язок між числом обчислювальних кроків орієнтованого ациклічного графа і числом перенесень одиниці до старшого розряду однозначно визначає мінімально достатнього числа перенесень для операції додавання бінарних кодів у схемі паралельного суматора з паралельним способом перенесення.

**Ключові слова:** ациклічний суматор з логічними елементами OR в останньому розряді, ациклічний граф, Kogge-Stone Adder, Han-Carlson Adder

**Замрій Б.А.** Исследование диапазона чисел ациклической модели сумматора с логическими элементами OR и XOR в последнем разряде. Проведенными исследованиями установлена перспектива увеличения производительности вычислительных сигналов суммы и переноса ациклических сумматоров бинарных кодов с логическими элементами OR в последнем разряде. Связь между числом вычислительных шагов ориентированного ациклического графа и числом переноса единицы к старшему разряду однозначно определяет минимально достаточное число переносов для операции суммирования бинарных кодов в схеме паралельного сумматора с паралельным способом переноса

**Ключевые слова:** ациклический сумматор с логическими элементами OR в последнем разряде, ациклический граф, Kogge-Stone Adder, Han-Carlson Adder

**Zamriy B.A.** Investigation of the range of numbers of the acyclic model compiler with logical OR and XOR elements in the last digit. The conducted researches have established the prospect of increasing the productivity of calculating the sum signals and transferring acyclic combiners of binary codes with OR logic elements in the last digit. The relationship between the number of computational steps of an oriented acyclic graph and the number of unit transitions to the senior level uniquely determines the minimum number of transitions for the operation of adding binary codes in the scheme of a parallel adder with a parallel transfer method.

**Keywords:** acyclic adder with logical elements OR in the last digit, acyclic graph, Kogge-Stone Adder, Han-Carlson Adder

### Вступ.

Ефективність додавання бінарних кодів суттєво залежить від дизайну суматора та методології обчислення сигналів суми і перенесення.

Бінарне додавання є головною арифметичною операцією у системах надвеликих інтегральних схем (НВІС). Двійкові суматори є одними з найважливіших елементів у процесорних мікросхемах, ALU, лічильниках, способах адресації пам'яті, є як частина фільтра, наприклад, фільтр DSP-решітки та ін. Структура суматора з послідовним перенесенням є однією з перших та найбільш фундаментальною для виконання операції бінарного додавання. Її швидкодія залежить від числа вхідних операндів, і, отже, затримка сигналу збільшується зі зростанням їхньої кількості. Паралельні префіксні суматори (PPA) [1–4] забезпечують кращу швидкодію, порівняно з суматорами з послідовним способом перенесення. Крім цього, будь-яке зменшення затримки безпосередньо стосується збільшення пропускну здатності [5].

У нанометровому діапазоні актуальним є розробка алгоритму додавання, реалізація якого використовує малу площу чипа, низьку потужність споживання та високу продуктивність його роботи.

Математичний апарат направлено ациклічного графа (рис. 1, 2) дозволяє однозначно отримати значення сигналів суми і перенесення за один етап обробки цифрових сигналів [6], тому останній спроможний з ефектом замінити трьох етапну префіксну модель обчислення.

Методи арифметичних операцій реалізуються вентильними схемами з функціональних елементів у базисах, що складаються з функцій алгебри логіки. Від структури суматора залежить швидкодія цифрового пристрою його надійність та енергозбереження. У зв'язку з цим мінімізація складності та глибини логічних схем є однією з центральних і практично важливих проблем у цій теорії, яка постає під час проектування цифрових пристроїв.

Процесорна еволюція є результатом невпинної оптимізації, тому актуальними залишаються дослідження направлені, зокрема, на вдосконалення таких чинників як:

- технології виготовлення;
- структурної реалізації;

- швидкодії та енергоспоживання;
- вартості цифрових пристроїв.

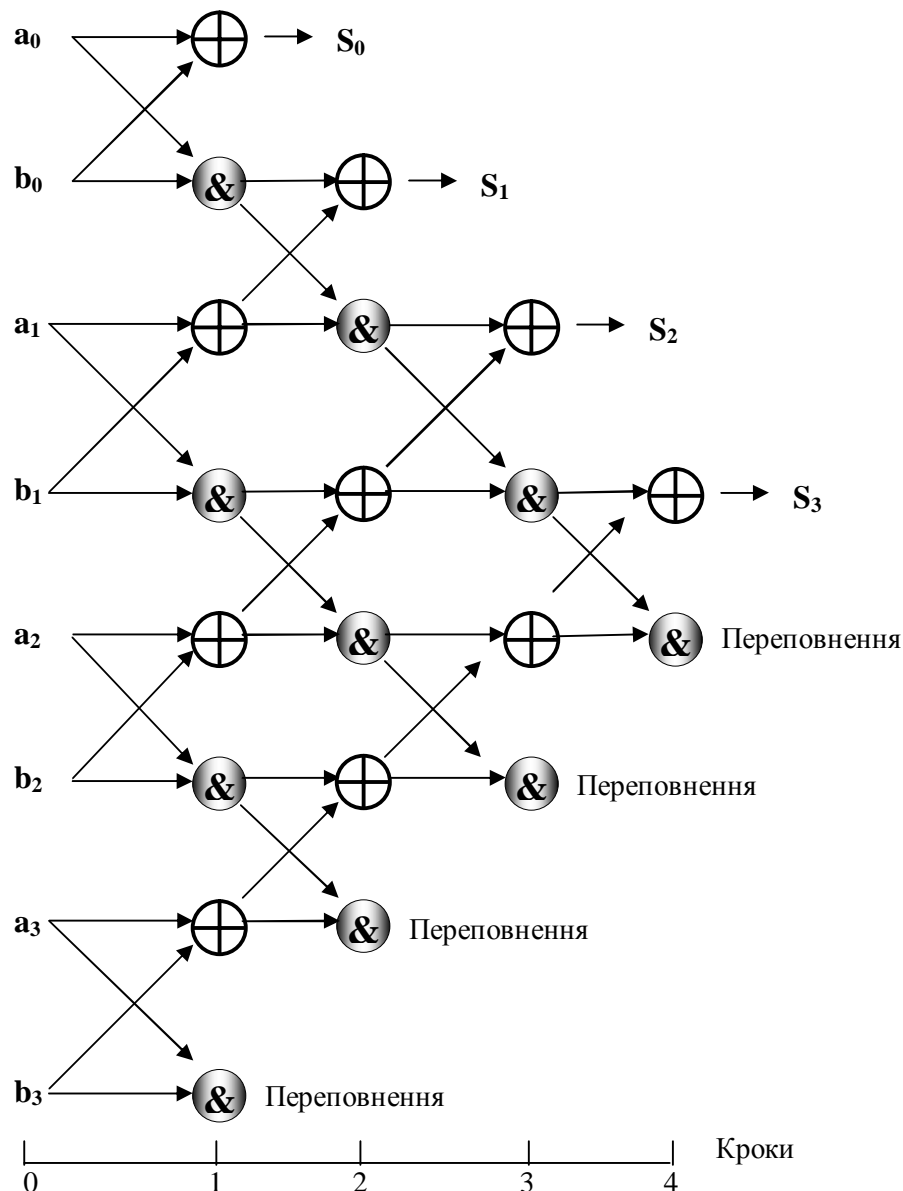


Рис. 1. Орієнтований ациклічний граф – модель обчислювальної схеми паралельного 4-розрядного ациклічного суматора з паралельним способом перенесення

#### Дослідження існуючих рішень проблеми.

Конфігурування обчислювальної структури у FPGA при якій суматори паралельного префікса мають кращу продуктивність представлено у роботі [7]. Паралельне розширення процесу обчислення є фундаментальною операцією в сучасних цифрових схемах та життєво важливою у великій частині комп'ютеризованих технологій, включаючи блоки ALU, мікрочіпи та розробку DSP. У зв'язку з цим у [7] проведено дослідження Kogge-Stone Adder (KSA), а також додаткові дослідження Ripple Carry Adder, Carry Look Adder та Carry Select Adder. Зазначається, що суматор Kogge-Stone є найшвидшим серед паралельно-префіксних суматорів, однак такий суматор має високу складність і величезну кількість з'єднувальних проводів. У роботі [7] продемонстровано ознаки поліпшення процесу обчислення за допомогою оборотних Gates (RLG). Оборотні схеми, які контролюють дані, шляхом розбивання бітів, на відміну від їх відкидання, скоро запропонують основний фізично можливий підхід для продовження збільшення продуктивності виконання обчислень.

Для зменшення складності Kogge-Stone Adder у роботі [8] запропонований так названий майже правильний суматор (суматор змінних затримок). Суматор змінних затримок, який заснований на паралельно-префіксній топології Han-Carlson, використовує спекуляцію: точна арифметична функція замінюється на апроксимовані значення, що швидше і дає правильний результат для більшості випадків, але не завжди. Для виявлення помилки використовується мережа виявлення помилок. Апроксимований суматор доповнюється мережею виявлення помилок, яка засвідчує сигнал помилки, коли спекуляція виходить з ладу. Спекулятивні суматори змінних затримок зменшують середню затримку у порівнянні з традиційними архітектурами. За допомогою Xilinx 14.3 синтезовано кілька спекулятивних суматорів змінної латентності для різних довжин операндів, що використовують топологію Han-Carlson та Kogge-Stone. Отримані результати показують, що запропонована змінна затримка Han-Carlson суматора використовується у високошвидкісному застосуванні.

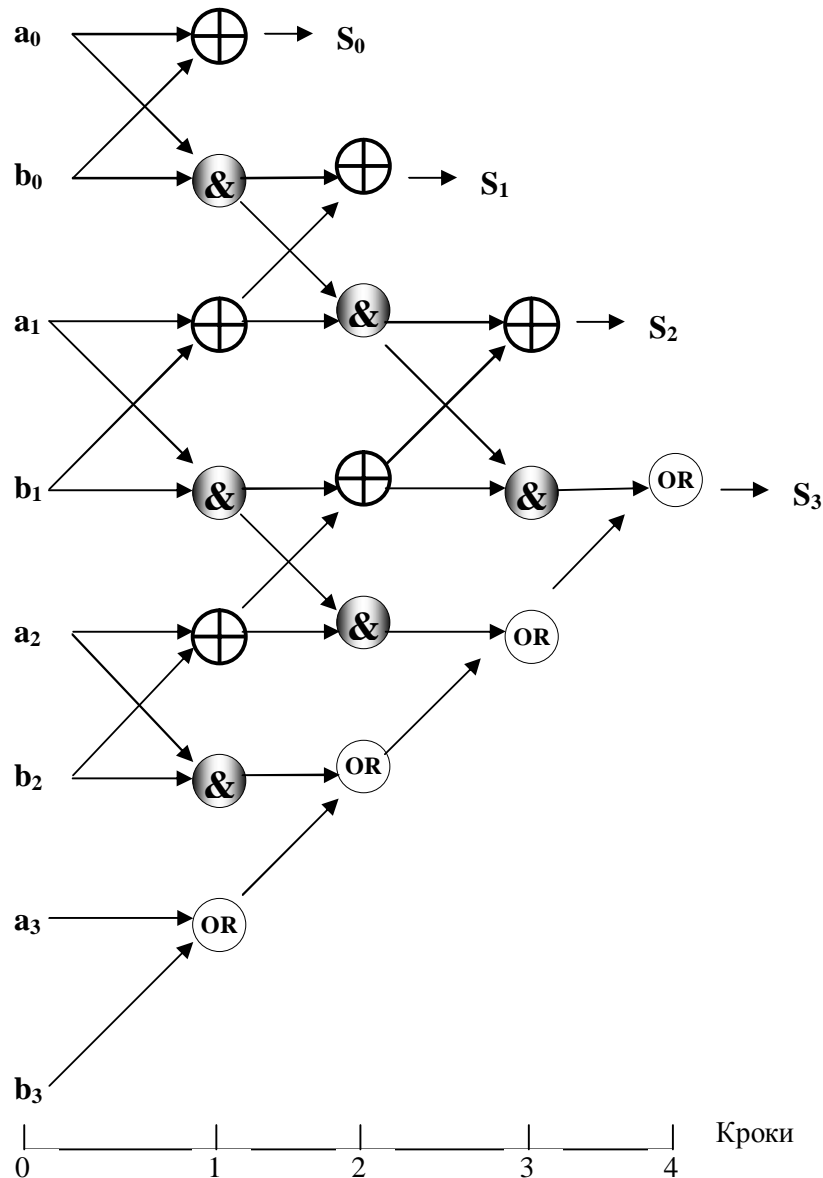


Рис. 2. Орієнтований ациклічний граф – модель обчислювальної схеми паралельного 4-розрядного ациклічного суматора з логічними елементами OR в останньому розряді

Незважаючи на те, що суматор Kogge-Stone і Han-Carlson є ефективними у тій чи іншій мірі, вони не можуть бути використані для входів з більшою кількістю бітів, оскільки вони, як кажуть, споживають більше місця, коли кількість вхідних бітів збільшується. Крім цього, збільшується і споживання енергії. У зв'язку з цим у роботі [9] представлена методика паралельного префікса, що застосовується для розробки ефективних суматорів, результати обчислення в яких визначаються

одним тактом синхронного імпульсу. Це сприяє зменшенню загальної площі чипа та загальної затримки без шкоди для таких параметрів, як продуктивність та енергоспоживання. Суматори, що розробляються використовують техніку QCA (Quantum-dot Cellular Automata - квантові точки клітинних автоматів), яка інтенсивно використовується для подальшого вдосконалення. Для спостереження за роботою суматора використовуються різні схеми синхронізації.

Каскадну схему, як механізм обчислення у складі префіксної моделі суматора, що використовує логічну структуру трьох етапного обчислення сигналів суми і перенесення, представлено у [10]. Зазначимо, що ациклічна модель обчислення сигналів суми і перенесення (рис. 1, 2) розрахована на логічну структуру суматора з послідовно-паралельним способом обчислення префікса та використовує структуру одно етапного обчислення. Таким чином, префіксна і ациклічна моделі є різними об'єктами – мають різні початки (принципи) обчислення, а від так володіють різними можливостями стосовно швидкодії обчислення, площі чипа та енергозбереження.

Конструкція суматорів, реалізованих з мемристорами представлена у роботі [11]. Тут роз'яснюються конструкції на основі мемристорів для стандартних архітектур суматорів (ripple carry adder, carry lookahead adder and parallel prefix adders). Порівнюються площі та затримки. Зазначено, що CLA має схожу складність з паралельними префіксними суматорами. Показано, що конструкція Kogge-Stone має кращу метрику з точки зору затримки і площі між паралельними суматорами префіксів.

Нова методологія проектування нечітких суматорів для прискорювачів обробки зображень розглядається у статті [12]. Запропонована методологія, зокрема, використовує архітектуру паралельного префікса та методи забезпечення низької потужності споживання за рахунок нечітких суматорів. Розглянуто два приклади для оцінки запропонованої методології: 1) фільтр гауссових зображень та 2) оператор Собеля. Результати показані на 45-нм технології де зниження споживання енергії коливається від 7,7% до 73,2% для декількох рівнів якості зображення.

На відміну від розглянутих публікацій (7 – 12), у даній роботі об'єктом аналізу методів збільшення продуктивності обчислень та зменшення складності цифрових компонентів є ациклічна модель обробки цифрових сигналів.

### **Мета та задачі дослідження**

Метою роботи є синтез оптимальних паралельних суматорів бінарних кодів з логічними елементами OR в останньому розряді, що забезпечують діапазон додавання чисел у межах від 0 до  $2^n - 1$ .

Для досягнення поставленої мети необхідно вирішити такі задачі:

1. Встановити адекватність математичної моделі на основі орієнтованого ациклічного графа з двома логічними операціями AND та XOR для синтезу суматорів бінарних кодів.
2. Оцінити динаміку збільшення глибини схеми паралельного ациклічного суматора з логічними елементами OR в останньому розряді зі збільшенням розрядної сітки схеми.
3. Встановити діапазон чисел ациклічного суматора з логічними елементами OR та XOR в останньому розряді.

### **Виклад основного матеріалу та обговорення отриманих результатів дослідження**

#### **Ациклічна модель обробки цифрових сигналів**

Число обчислювальних кроків орієнтованого ациклічного графа з двома логічними операціями AND і XOR (рис. 1, 2) визначає оптимальне число перенесень у схемі  $n$ -bit паралельного суматора бінарних кодів [13]. Наприклад, вісім обчислювальних кроків орієнтованого ациклічного графа визначає вісім перенесень у схемі 8-bit PAA. Зазначене співвідношення виконується тільки для 4- та 8-bit суматорів. Зі збільшенням розрядності ациклічного суматора (16-, 32-, 64-bit ...) число обчислювальних кроків визначається за логарифмічним законом (рис. 3).

Динаміка збільшення глибини схеми PAA визначається логарифмічною залежністю – подвоєння розрядності  $n$  суматора збільшує глибину схеми на сталу величину – на два логічних елементи.

У випадку, коли синтезований суматор отримав більше число перенесень порівняно з числом обчислювальних кроків відповідного орієнтованого ациклічного графа, то такий суматор буде неоптимальним стосовно числа обчислювальних операцій.

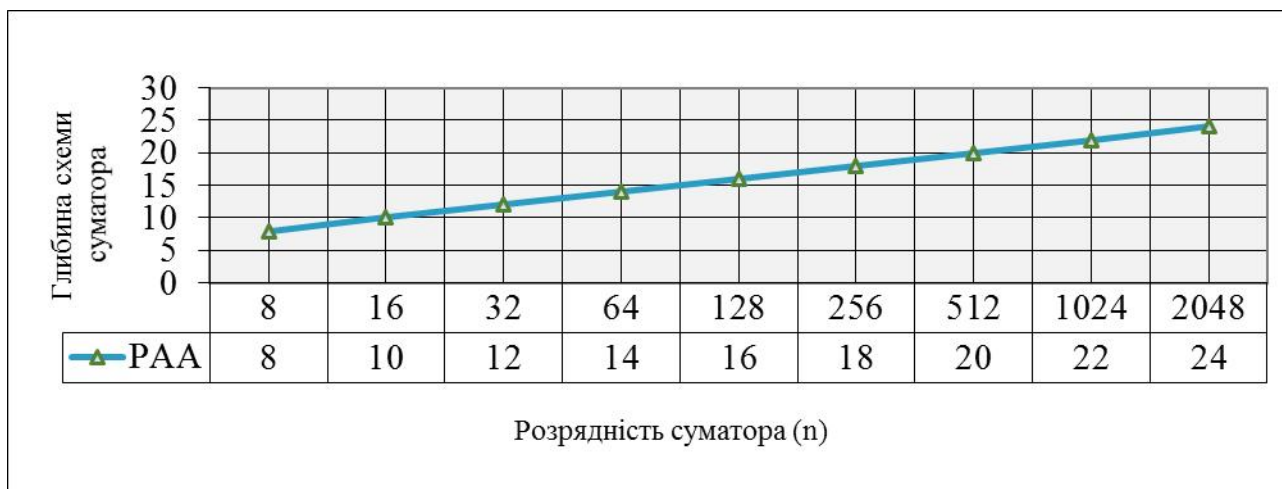


Рис. 3. Динаміка збільшення глибини схеми паралельного ациклічного суматора (РАА)

Основними недоліками префіксної моделі обчислення сигналів суми і перенесення є:

- організація процесу паралельного обчислення префікса передбачає початок обчислення з першого розряду схеми суматора (це й є власне шлях (метод) префікса), що приводить, у підсумку, до надлишкового нагромадження та ускладнення апаратної частини пристрою [14];

- принцип три етапного вироблення сигналу суми і перенесення, що задає певну складність такого обчислення, зокрема ускладнює дидактику методу;

У свою чергу застосування ациклічної моделі розраховано на:

- процес послідовного (для молодших розрядів схеми суматора) та паралельного (для решти розрядів) обчислення сигналів суми і перенесення, що, у підсумку, дає зменшення складності апаратної частини пристрою та не збільшує глибину схеми;

- встановлення оптимального числа обчислювальних кроків.

Отже, використання ациклічної моделі, порівняно з префіксною моделлю, для синтезу схем суматорів бінарних кодів, дозволяє збільшити продуктивність обчислень та зменшити енерговитрати цифровими компонентами.

#### Діапазон чисел ациклічної моделі

Результат виконання операції додавання над бітами  $a_i$  і  $b_i$  у  $i$ -му розряді бінарного коду виражається двома параметрами:  $c_i$  – результатом операції додавання бітів поточного розряду бінарного коду та цифрою  $p_{i+1}$  – перенесенням одиниці до старшого розряду. Результати порозрядного виконання операцій  $c_i$  та перенесення  $p_{i+1}$  формуються за правилами (1).

Для ациклічної моделі на рис. 1 (з логічними елементами XOR в останньому розряді) правила (1) будуть виконуватись. Це забезпечить діапазон чисел у бінарному коді для  $n$ -розрядної сітки у межах від 0 до  $2^n - 1$ . Наприклад для 8-розрядної сітки діапазон чисел у бінарному коді ациклічної моделі на рис. 1 складе від 0 до 255.

$$c_i = \begin{cases} a_i + b_i, & \text{при } a_i + b_i < q; \\ a_i + b_i - q, & \text{при } a_i + b_i > q. \end{cases} \quad (1)$$

$$p_{i+1} = \begin{cases} 0, & \text{при } a_i + b_i < q; \\ 1, & \text{при } a_i + b_i > q. \end{cases}$$

Значимо, що число всіх пар  $n$ -біт аргументів  $N$ , які можуть прийняти участь в операції додавання становить

$$N = 2^n \times 2^n = 2^{2n}.$$

Наприклад, для 4-bit аргументів число пар  $N = 256$ . З них 136 пар забезпечують діапазон додавання чисел у бінарному коді для 4-розрядної сітки у межах від 0 до  $2^4 - 1$ . Решта пар дадуть переповнення розрядної сітки схеми суматора (рис. 4).

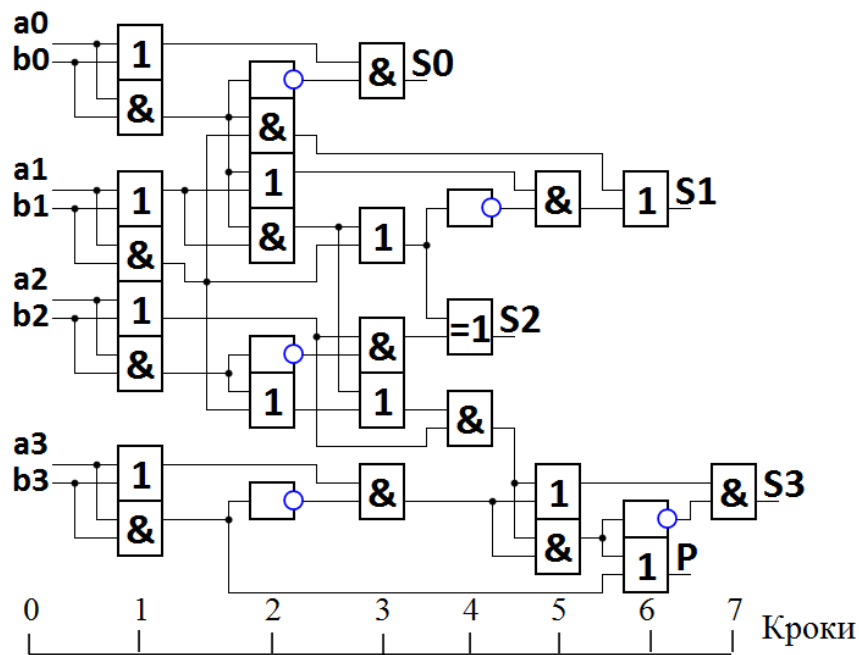


Рис. 4. 4-bit ациклічний суматор з елементами XOR в останньому розряді  
 Логічні рівняння 4-bit ациклічного суматора на рис. 4 є такі:

$$\begin{aligned}
 S_0 &= a_0 \bar{b}_0 + \bar{a}_0 b_0; \\
 S_1 &= a_0 b_0 \bar{a}_1 \bar{b}_1 + \bar{a}_0 a_1 \bar{b}_1 + \bar{a}_0 \bar{a}_1 b_1 + a_0 b_0 a_1 b_1 + \bar{b}_0 a_1 \bar{b}_1 + \bar{b}_0 a_1 b_1; \\
 S_2 &= \bar{a}_0 \bar{b}_1 a_2 \bar{b}_2 + \bar{a}_0 a_1 a_2 \bar{b}_2 + \bar{a}_0 b_1 a_2 b_2 + \bar{a}_0 \bar{a}_1 a_2 b_2 + \bar{b}_0 \bar{b}_1 a_2 \bar{b}_2 + \\
 &\quad + \bar{b}_0 a_1 a_2 \bar{b}_2 + \bar{b}_0 b_1 a_2 b_2 + \bar{b}_0 \bar{a}_1 a_2 b_2 + a_0 b_0 \bar{b}_1 a_2 \bar{b}_2 + \\
 &\quad + a_0 b_0 a_1 a_2 \bar{b}_2 + a_0 b_0 b_1 a_2 b_2 + a_0 b_0 a_1 a_2 b_2 + \bar{a}_1 \bar{b}_1 a_2 \bar{b}_2 + \\
 &\quad + \bar{a}_1 \bar{b}_1 a_2 b_2 + \bar{a}_1 b_1 a_2 \bar{b}_2 + \bar{a}_1 b_1 a_2 b_2; \\
 S_3 &= \bar{a}_0 \bar{b}_1 \bar{b}_2 a_3 \bar{b}_3 + \bar{a}_0 a_1 \bar{b}_2 a_3 \bar{b}_3 + \bar{a}_0 \bar{b}_1 a_2 a_3 \bar{b}_3 + \bar{a}_0 a_1 a_2 a_3 \bar{b}_3 + \\
 &\quad + \bar{a}_0 \bar{b}_1 b_2 a_3 \bar{b}_3 + \bar{a}_0 a_1 b_2 a_3 \bar{b}_3 + \bar{a}_0 \bar{b}_1 a_2 a_3 b_3 + \bar{a}_0 a_1 a_2 a_3 b_3 + \\
 &\quad + \bar{b}_0 \bar{b}_1 \bar{b}_2 a_3 \bar{b}_3 + \bar{b}_0 a_1 \bar{b}_2 a_3 \bar{b}_3 + \bar{b}_0 \bar{b}_1 a_2 a_3 \bar{b}_3 + \bar{b}_0 a_1 a_2 a_3 \bar{b}_3 + \\
 &\quad + \bar{b}_0 \bar{b}_1 b_2 a_3 \bar{b}_3 + \bar{b}_0 a_1 b_2 a_3 \bar{b}_3 + \bar{b}_0 \bar{b}_1 a_2 a_3 b_3 + \bar{b}_0 a_1 a_2 a_3 b_3 + \\
 &\quad + \bar{a}_1 \bar{b}_1 \bar{b}_2 a_3 \bar{b}_3 + \bar{a}_1 \bar{b}_1 a_2 a_3 \bar{b}_3 + \bar{a}_1 \bar{b}_1 b_2 a_3 \bar{b}_3 + \bar{a}_1 \bar{b}_1 a_2 a_3 b_3 + \\
 &\quad + a_0 b_0 \bar{b}_1 \bar{b}_2 a_3 \bar{b}_3 + a_0 b_0 a_1 \bar{b}_2 a_3 \bar{b}_3 + a_0 b_0 \bar{b}_1 a_2 a_3 \bar{b}_3 + \\
 &\quad + a_0 b_0 a_1 b_2 a_3 \bar{b}_3 + a_0 b_0 b_1 b_2 a_3 \bar{b}_3 + a_0 b_0 a_1 b_2 a_3 b_3 + \\
 &\quad + a_0 b_0 \bar{b}_1 a_2 a_3 \bar{b}_3 + a_0 b_0 a_1 a_2 a_3 \bar{b}_3 + \bar{a}_1 \bar{b}_1 \bar{b}_2 a_3 \bar{b}_3 + \bar{a}_1 \bar{b}_1 a_2 a_3 \bar{b}_3 + \\
 &\quad + \bar{a}_1 \bar{b}_1 b_2 a_3 \bar{b}_3 + \bar{a}_1 \bar{b}_1 a_2 a_3 b_3 + \bar{a}_2 \bar{b}_2 a_3 \bar{b}_3 + \bar{a}_2 \bar{b}_2 a_3 b_3 + \\
 &\quad + \bar{a}_2 b_2 a_3 \bar{b}_3 + \bar{a}_2 b_2 a_3 b_3;
 \end{aligned}$$

$$\begin{aligned}
 P = & a_0 b_0 b_1 b_2 b_3 + a_0 b_0 a_1 b_2 b_3 + a_0 b_0 b_1 a_2 b_3 + a_0 b_0 a_1 a_2 b_3 + \\
 & + a_0 b_0 b_1 b_2 a_3 + a_0 b_0 a_1 b_2 a_3 + a_0 b_0 b_1 a_2 a_3 + a_0 b_0 a_1 a_2 a_3 + \\
 & + a_1 b_1 b_2 b_3 + a_1 b_1 a_2 b_3 + a_1 b_1 b_2 a_3 + a_1 b_1 a_2 a_3 + a_2 b_2 b_3 + \\
 & + a_2 b_2 a_3 + a_3 b_3.
 \end{aligned}$$

Для ациклічної моделі на рис. 2 (з логічними елементами OR в останньому розряді) правила (1) не виконуються в останньому розряді додавання бінарних кодів. Однак логіка ациклічної моделі на рис. 2 при не виконанні правила (1) в останньому розряді забезпечує переповнення розрядної сітки схеми суматора (рис. 5).

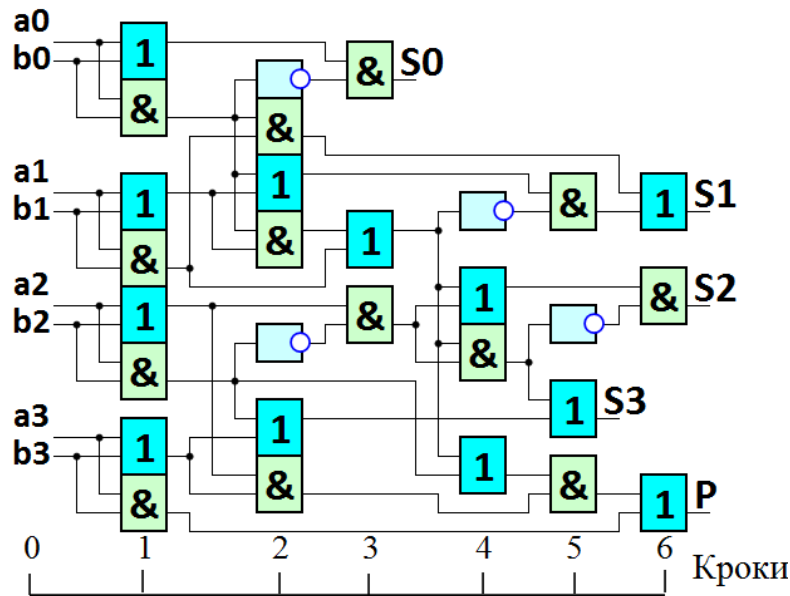


Рис. 5. 4-bit ациклічний суматор з елементами OR в останньому розряді

Таким чином, не виконання правила (1) в останньому розряді ациклічної моделі суматора фіксується сигналом переповнення розрядної сітки. При цьому 136 пар 4-bit аргументів забезпечують діапазон додавання чисел ациклічного суматора з логічними елементами OR в останньому розряді у межах від 0 до  $2^4 - 1$ . У загальному випадку у межах від 0 до  $2^n - 1$ . Решта пар дадуть переповнення у бінарному коді для  $n$ -розрядної сітки схеми суматора.

Логічні рівняння 4-bit ациклічного суматора на рис. 5 є такі:

$$\begin{aligned}
 S_0 &= a_0 \bar{b}_0 + \bar{a}_0 b_0; \\
 S_1 &= a_0 \bar{b}_0 a_1 \bar{b}_1 + \bar{a}_0 a_1 \bar{b}_1 + \bar{a}_0 a_1 b_1 + b_0 a_1 \bar{b}_1 + \bar{b}_0 a_1 b_1 + a_0 b_0 a_1 b_1; \\
 S_2 &= \bar{a}_0 \bar{b}_1 a_2 \bar{b}_2 + \bar{a}_0 a_1 a_2 \bar{b}_2 + \bar{a}_0 \bar{b}_1 a_2 b_2 + \bar{a}_0 a_1 a_2 b_2 + \bar{b}_0 \bar{b}_1 a_2 \bar{b}_2 + \\
 &+ \bar{b}_0 a_1 a_2 \bar{b}_2 + \bar{b}_0 \bar{b}_1 a_2 b_2 + \bar{b}_0 a_1 a_2 b_2 + a_0 b_0 \bar{b}_1 a_2 \bar{b}_2 + \\
 &+ a_0 b_0 a_1 a_2 \bar{b}_2 + a_0 b_0 \bar{b}_1 a_2 b_2 + a_0 b_0 a_1 a_2 b_2 + \\
 &+ a_1 \bar{b}_1 a_2 \bar{b}_2 + a_1 b_1 a_2 b_2; \\
 S_3 &= +a_0 b_0 b_1 b_2 + a_0 b_0 a_1 b_2 + a_0 b_0 b_1 a_2 + a_0 b_0 a_1 a_2 + a_1 b_1 b_2 + \\
 &+ a_1 b_1 a_2 + a_2 b_2 + a_3 + b_3;
 \end{aligned}$$

$$P = a_0b_0b_1b_2b_3 + a_0b_0a_1b_2b_3 + a_0b_0b_1a_2b_3 + a_0b_0a_1a_2b_3 + \\
 + a_0b_0b_1b_2a_3 + a_0b_0a_1b_2a_3 + a_0b_0b_1a_2a_3 + a_0b_0a_1a_2a_3 + \\
 + a_1b_1b_2b_3 + a_1b_1a_2b_3 + a_1b_1b_2a_3 + a_1b_1a_2a_3 + a_2b_2b_3 + \\
 + a_2b_2a_3 + a_3b_3.$$

Таблиця 1

Порівняльна таблиця параметрів  
 4-bit ациклічних суматорів з елементами OR та XOR в останньому розряді

Параметри	4-bit суматор з логічними елементами OR в останньому розряді	4-bit суматор з логічними елементами XOR в останньому розряді
Складність схеми	29	33
Глибина схеми	6	7
Діапазон додавання чисел	$2^4 - 1$	$2^4 - 1$
Число всіх пар 4-bit аргументів, які можуть приймати участь в операції додавання	256	256
Число пар 4-bit аргументів, що забезпечують роботу суматора без переповнення	136	136
Відсоток пар 4-bit аргументів, що забезпечують роботу суматора без переповнення	53,13%	53,13%

Споглядаючи табл. 1 бачимо, що обидва суматори, з логічними елементами OR та XOR, забезпечують однаковий діапазон додавання чисел, у межах від 0 до  $2^4 - 1$ . Однак суматор з логічними елементами OR в останньому розряді є швидшим (глибина схеми 6 елементів) та має простішу структуру (складність схеми 29 елементів), порівняно зі схемою суматора з логічними елементами XOR в останньому розряді.

**Обговорення результатів застосування ациклічної моделі для синтезу суматорів бінарних кодів з логічними елементами OR в останньому розряді**

Застосування ациклічної моделі розраховано на:

- процес послідовного (для молодших розрядів схеми суматора) та паралельного (для решти розрядів) обчислення сигналів суми і перенесення, що, у підсумку, дає зменшення складності апаратної частини пристрою та не збільшує глибину схеми;
- встановлення оптимального числа обчислювальних кроків.

Це вказує на те, що обчислювальні кроки орієнтованого ациклічного графа і перенесення одиниці до старшого розряду суматора являють собою один об'єкт. Наприклад, вісім обчислювальних кроків орієнтованого ациклічного графа визначає вісім перенесень у схемі 8-bit РАА. Зазначене співвідношення виконується тільки для 4- та 8-bit суматорів. Зі збільшенням розрядності ациклічного суматора (16-, 32-, 64-bit ...) число обчислювальних кроків визначається за логарифмічним законом.

Ациклічні суматори, з логічними елементами OR та XOR, забезпечують однаковий діапазон додавання чисел, у межах від 0 до  $2^n - 1$ . Однак суматор з логічними елементами OR в останньому розряді є швидшим (глибина схеми 6 елементів) та має простішу структуру (складність схеми 29 елементів), порівняно зі схемою суматора з логічними елементами XOR в останньому розряді.

Отже, використання ациклічної моделі для синтезу паралельних суматорів з логічними елементами OR в останньому розряді, дозволяє збільшити продуктивність обчислень та зменшити енерговитрати цифровими компонентами.

Використання ациклічної моделі вигідніше у порівнянні з аналогами за такими чинниками:



– меншою вартістю розробки та впровадження, оскільки ациклічна модель визначає порівняно простішу структуру суматора;

– наявністю критерію оптимізації – число обчислювальних кроків ациклічного графа вказує на мінімально достатнє число перенесень одиниці до старшого розряду.

Оскільки ациклічна модель демонструє 4-bit PAA з глибиною схеми 6 типових 2-входових елементів при використанні логічних елементів OR в останньому розряді (рис. 5) і 4-bit PAA з глибиною схеми 7 типових 2-входових елементів при використанні логічних елементів XOR в останньому розряді (рис. 7), перспективою подальших досліджень цифрових схем може бути переоцінка методу паралельного розширення процесу обчислення в сучасних цифрових пристроях, переоцінка алгоритмів додавання у нанометровому діапазоні, переоцінка конструкції суматорів, реалізованих з мемристорами та ін.

## Висновки

1. Встановлено, що обчислення сигналу суми і перенесення у схемі паралельного ациклічного суматора здійснюється за алгоритмом логарифмічного додавання. Число обчислювальних кроків ациклічного графа визначає оптимальне число перенесень у схемі паралельного суматора з паралельним способом перенесення.

2. Оцінка динаміки збільшення глибини схеми ациклічного суматора з логічними елементами OR в останньому розряді складає  $O(n)$  і є лінійною для  $n \leq 8$ . Зі збільшенням розрядності схеми від  $n > 8$  оцінка динаміки збільшення глибини схеми ациклічного суматора складає  $O(\log n)$  і є логарифмічною.

3. Суматори, з логічними елементами OR та XOR, забезпечують однаковий діапазон додавання чисел, у межах від 0 до  $2^n - 1$ . Однак суматор з логічними елементами OR в останньому розряді є швидшим (глибина схеми 6 елементів) (рис. 5) та має простішу структуру (складність схеми 29 елементів), порівняно зі схемою суматора з логічними елементами XOR в останньому розряді (рис. 4).

З огляду на зазначене порівняння ациклічних суматорів з елементами OR та XOR в останньому розряді, структура суматора з логічними елементами OR в останньому розряді дає підставу для доцільності її застосування у процесах синтезу арифметичних пристроїв обробки цифрових даних, оскільки зазначена схема суматора спроможна:

– збільшити швидкодію;

– зменшити енергоспоживання та тепловиділення цифрового пристрою, інтегральної схеми.

1. Brent R. P., Kung H. T. A regular layout for parallel adders // IEEE Tr. Comp., C-31(3): Mar. 1982, pp. 260–264.

2. Han T., Carlson D. A. Fast area-efficient VLSI adders // In 8th Symp. on Comp. Arithmetic, May 1987.

3. Kogge P., Stone H. A parallel algorithm for the efficient solution of a general class of recurrence equations // IEEE Tr. Comp., C-22(8): Aug. 1973, pp. 786–793.

4. Ladner R. E., Fischer M. J. Parallel prefix computation // Journal of the ACM, 27(4): Oct. 1980. pp. 831–838.

5. Choi, Y. Parallel Prefix Adder Design with Matrix Representation // Proc. 17th IEEE Symposium on Computer Arithmetic, 27th. June 2005, pp 90-98.

6. Solomko M., Olshansky P. The Parallel Acyclic Adder // 2017 14<sup>th</sup> International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), Lviv, 2017, pp. 125-129.

7. Srinivasarao B.N., Prathyusha Ch. Power Efficient Parallel Prefix Adders // International Journal of Research, 2018, February. pp. 472-477. URL:

[http://scholar.google.com.ua/scholar\\_url?url=https://pen2print.org/index.php/jjr/article/download/12158/11483&hl=uk&sa=X&d=7093867988941575373&scisig=AAGBfm2PGwrG0bBN99RZuzx6QcF2cRUpnA&nossl=1&oi=scholarart](http://scholar.google.com.ua/scholar_url?url=https://pen2print.org/index.php/jjr/article/download/12158/11483&hl=uk&sa=X&d=7093867988941575373&scisig=AAGBfm2PGwrG0bBN99RZuzx6QcF2cRUpnA&nossl=1&oi=scholarart)

8. Karthik K., Rajeshwar B. A New Design for Variable Latency Speculative EC & D Han-Carlson Adder // International Journal of Research, Volume 04, Issue 13. October 2017. pp. 975-980. URL:

[http://scholar.google.com.ua/scholar\\_url?url=https://pen2print.org/index.php/jjr/article/download/9332/8980&hl=uk&sa=X&d=5289417270700682252&scisig=AAGBfm2SRr4V6L-sliX7AvD4DsUamxYCSA&nossl=1&oi=scholarart](http://scholar.google.com.ua/scholar_url?url=https://pen2print.org/index.php/jjr/article/download/9332/8980&hl=uk&sa=X&d=5289417270700682252&scisig=AAGBfm2SRr4V6L-sliX7AvD4DsUamxYCSA&nossl=1&oi=scholarart)

9. Hima Bindu Challa, Srujana Gollapalli, Dr. Varaprasada Rao . M. Design-of-a-novel-BCD-adder-using-parallel-prefix technique // INTERNATIONAL JOURNAL OF RESEARCH IN ELECTRONICS AND COMPUTER ENGINEERING, VOL. 6, ISSUE 2. APR.-JUNE 2018. pp. 2213-2219. URL: [https://www.researchgate.net/profile/Srujana\\_Gollapalli/publication/329425015\\_Design\\_of\\_a\\_novel\\_BCD\\_adder\\_using\\_parallel\\_prefix\\_technique/links/5c07f505299bf139c741adb5/Design-of-a-novel-BCD-adder-using-parallel-prefix-technique.pdf](https://www.researchgate.net/profile/Srujana_Gollapalli/publication/329425015_Design_of_a_novel_BCD_adder_using_parallel_prefix_technique/links/5c07f505299bf139c741adb5/Design-of-a-novel-BCD-adder-using-parallel-prefix-technique.pdf)

10. Balasubramanian P., Jacob Prathap Raj C., Anandi S. Mathematical Modeling of Timing Attributes of Self-Timed Carry Select Adders // Recent Advances in Circuits, Systems, Telecommunications and Control. 2013. pp. 228-243. URL: <http://www.wseas.us/e-library/conferences/2013/Paris/CCTC/CCTC-34.pdf>

11. Nagaraja Revanna, Earl E. Swartzlander Memristor Adder Design // IEEE 61<sup>st</sup> International Midwest Symposium on Circuits and Systems (MWSCAS). 5-8 Aug. 2018. URL: <https://ieeexplore.ieee.org/abstract/document/8623864>

12. Soares L.B., Azevedo da Rosa M.M. Design Methodology to Explore Hybrid Approximate Adders for Energy-Efficient Image and Video Processing Accelerators // IEEE Transactions on Circuits and Systems I: Regular Papers ( Early Access ). 31 January 2019. pp. 1-14. URL:

<https://ieeexplore.ieee.org/abstract/document/8630652>

13. Solomko M., Krulikovskiy B. Study of carry optimization while adding binary numbers in the rademacher number-theoretic basis // Eastern-European Journal of Enterprise Technologies. Mathematics and Cybernetics – applied aspects, Volume 3, № 4(81). 2016. pp: 56-63. URL: <http://journals.uran.ua/eejet/article/view/70355>

14. Solomko M. Optimization of the acyclic adders of binary codes // Technology audit and production reserves, Vol 3/2 (41). 2018. pp 55 - 65. URL:

<http://journals.uran.ua/tarp/article/view/133694>