регістрів FPRs для першого операнда, RS2 – відповідно для другого операнда, RD – пару регістрів для результату виконання інтервальної інструкції.

При використанні операторів нечіткої арифметики в складі мов високого рівня кожний з них транслюється в такі послідовності машинних інструкцій:

1. Блок інструкцій завантаження нечітких операндів (у вигляді рівневих множин) з пам'яті до регістрового файла рухомої коми FPRs. Можуть бути використані такі інструкції: LF Fd, Adr або LD Dr, Adr (Load single/double precision floating point).

2. Блок інструкцій інтервальної арифметики (в межах регістрової структури процесора): IADDF, ISUBF, IMULF, IDIVF.

3. Блок інструкцій запису до пам'яті нечіткого результату (у вигляді рівневих множин) з регістрового файла рухомої коми FPRs, : SF Adr, Fs або SD Adr, Fs (Store single/double precision floating point).

Висновки. Оператори обробки нечітких даних добре векторизуються. Блокування інструкцій дозволяє зменшити ймовірність зривів конвеєра RISC-процесора через залежність даних або умовні переходи. Як правило, серійні процесори мають невикористані можливості для суперскалярного або LVIW виконання інструкцій, завдяки чому їх функції можна розширити з найменшими витратами.

1. Дюбуа Д., Прад А. Теория возможностей. Приложения к представлению знаний в информатике. М.,1990. 2. Борисов А.М., Крумбер О.А., Федоров И.П. Принятие решений на основе нечетких моделей. Примери использования. Рига, 1990. 3. Scott Ferson, I Alin Cooper, Dwayne R.I. Moore, Robert C. Lee. Beyond Point Estimates. Risk Assessment Using Interval, Fuzzy and Probabilistic Arithmetic. Society for Risc Analisys, Washington, DC, December1997. 4. Nguen H.T., Koshelev M., Kreinovich V., Kosheleva O. Computational Complexity and Feasibility of Fuzzy Data Procesing: Why Fuzzy Numbers, Wich Fuzzy Numbers, Wich Operations with Fuzzy Numbers. Technical Report UTEP-CS-97-28, November 1997. 5. David A. Patterson, John I. Mennesy. Computer Architecture. Aquantitative Approach. San-Francisco, California, 1996.

УДК 621.382.323

Бєляєв С. М., Троценко В. В.
ДУ "Львівська політехніка", кафедра ЕОМ

АПАРАТНО-ОРІЄНТОВАНА КОНТЕКСТНА ТЕХНОЛОГІЯ СТИСКУ ІНТЕНСИВНИХ ВІДЕОПОТОКІВ В РЕАЛЬНОМУ ЧАСІ

Розглянуто програмну модель апаратно-орієнтованої технології стиску монохромних растрових відеопотоків модифікованого варіанта відомого [1] алгоритму контекстно-базованої технології стиску з метою підвищення коефіцієнта стиску і пристосування до швидкої апаратної реалізації (з інтенсивністю ~100 мегапікселів на секунду). Надано порівняння результатів стиску монохромних зображень з іншою контекстно-базованою технологією [2], яка претендує на проголошення стандартом JPEG-2000.

Вступ. При створенні модифікованого варіанту нової технології стиску без втрат та відповідної їй програмної моделі за основу нами узято алгоритм [1]. Питання привнесення втрат розглядалося окремо і вирішувалося методом препроцесорного зсуву з втратою молодших розрядів відліків інтенсивностей кожного пікселя відеопотоку.

Опис програмної моделі. Запропонований варіант технології грунтується на гіпотезі про розподіл, яку репрезентує гістограма інтенсивностей сірого растрового зображення (рис.1). З цієї гіпотези випливає контекстна модель, яку використано при побудові алгоритму. Контекстна модель – це статистична модель, яка розглядає зображення як сукупність областей – контекстів, елементи яких є більш залежними один від одного порівняно із елементами зображення (пікселями) в цілому. В даній технології під контекстом ми розуміємо абсолютну величину різниці між інтенсивностями діагональних сусідів пікселя, що кодується. Кожен піксель може належати до однієї з трьох областей. При цьому предикція не використовується, потрібно лише оптимально закодувати значення інтенсивності даного пікселя. Для цього використовують бінарні коди змінної довжини для центральної зони та коди Голомба-Риса для бічних зон, а також поодинокі біти, що забезпечують префікси кодів та однозначну роботу схеми декодування.

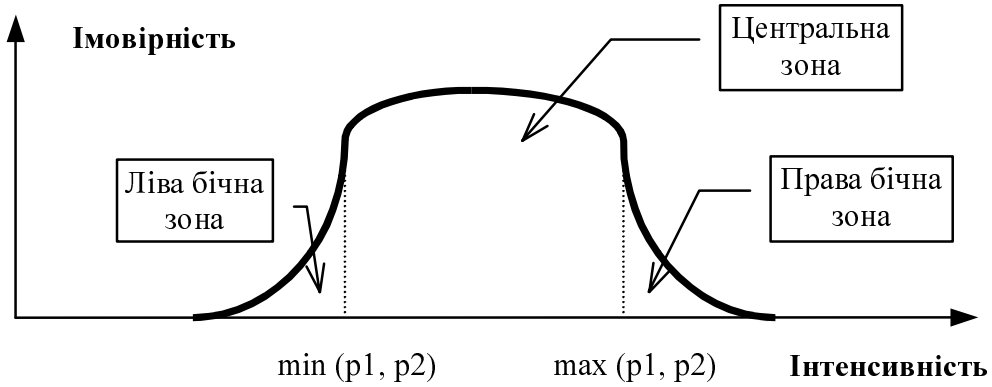

Рис. 1 Гістограма розподілу інтенсивностей пікселів растрового зображення

Бінарні коди змінної довжини фактично є кодами Хафмана із заданим розподілом імовірністей. Коди Голомба-Риса наведено в табл. 1. Використання кодів Голомба-Риса пояснюється тим, що крива розподілу імовірностей для цих кодів є подібною до кривої розподілу (рис. 1) бічних зон гістограми.

Таблиця 1

Коди Голомба-Риса

значення\параметр	k=0	k=1	k=2	k=3
0	0	0.0	0.00	0.000
1	10	0.1	0.01	0.001
2	110	10.0	0.10	0.010
3	1110	10.1	0.11	0.011
4	11110	110.0	10.00	10.100
...	...	...	...	...

Як видно з таблиці 1, коди Голомба-Риса залежать від параметра k. Під час роботи з програмною моделлю нами встановлено, що при стиску без втрат необхідно використовувати параметр k=0...3; для стиску з абсолютною похибкою $\Delta = \pm 1$ k=0...2; для $\Delta = \pm 2, \pm 4$ – k=0,1. Це має силу на визначеній нами множині проблемних та мультимедійних зображень.

Сутність запропонованої нами модифікації технології полягає у збільшенні коефіцієнта стиску без суттєвого ускладнення алгоритму. Враховуючи, що коди, які застосовують, є кодами змінної довжини, очевидним шляхом підвищення коефіцієнту стиску є зменшення кількості бітів на піксель в коді. Експериментально визначено, що для розподілу (рис. 1) близько 50% пікселів розташовано в центральній області і по 25% в бічних. Отже, аби забезпечити ефективність, необхідно зменшити діапазон значень кода, який чисельно дорівнює номеру контекста.

У зображеннях природнього походження значення інтенсивності сусідніх пікселів зазвичай відрізняється від даного не більше, ніж на декілька одиниць. Для цих сусідів характерний експоненційний розподіл. Відомо, що випадкова величина ξ має розподіл Лапласа (двобічний експоненційний розподіл), якщо

$$\xi = \xi_1 - \xi_2 ,$$

де ξ_1, ξ_2 – випадкові величини, що мають експоненційний розподіл [3].

Отже, величина контексту має експоненційний розподіл, оскільки дорівнює модулю від різниці двох випадкових величин з експоненційним розподілом. Тобто більшість пікселів потрапляє до контекстів з меншим номером. При роботі з програмною моделлю виявлено, що при значеннях контексту більших ніж 2 можна підвищити коефіцієнт стиску в середньому на 1-3%, якщо не включати межі (рис. 1) min(p1,p2) та max(p1,p2) до центральної області. Результати стиску, отримані при цьому, надано в колонці “Модифікована контекстно-базована технологія [1]”.

Результати, які отримано на проблемних та мультимедійних тестових зображеннях, наведено в таблицях 2-5.

Таблиця 2

Результати стиску проблемних та мультимедійних зображень без втрат

Зображення	Контекстно-базована технологія [1]	Апаратно-орієнтована контекстна технологія [1]	Контекстно-базована технологія [2]
problem1	1,90	1,92	2,07
problem2	2,12	2,15	2,42
problem3	2,10	2,12	2,19
problem4	2,15	2,17	2,25
problem5	2,19	2,23	2,38
problem6	2,26	2,30	2,38
problem7	1,89	1,91	2,01
problem8	1,70	1,70	1,81
problem9	1,55	1,55	1,65
problem10	1,84	1,84	1,93
boat	1,71	1,71	1,88
collie	2,12	2,15	2,75
lenna	1,78	1,80	1,94
san256	2,13	2,15	2,72

Таблиця 3

Результати стиску проблемних та мультимедійних зображень з втратами (абсолютна похибка не більша ніж 1).

Зображення	Ср. кв. відх.	Контекстно-базована технологія [1]	Апаратно-орієнтована контекстна технологія [1]	Ср. кв. відх.	Контекстно-базована технологія [2]
1	2	3	4	5	6
problem1	0,71	2,48	2,52	0,81	3,03
problem2	0,71	2,80	2,89	0,81	3,83
problem3	0,71	2,78	2,86	0,81	3,72
problem4	0,71	2,84	2,92	0,81	3,82
problem5	0,71	2,84	2,90	0,81	3,98
problem6	0,71	2,97	3,05	0,81	4,14
problem7	0,71	2,47	2,52	0,81	3,20
problem8	0,71	2,16	2,17	0,81	2,74
problem9	0,71	1,93	1,94	0,81	2,39
problem10	0,71	2,38	2,42	0,81	3,02
boat	0,71	2,19	2,18	0,81	2,91
collie	0,71	2,79	2,87	0,81	4,55
lenna	0,71	2,31	2,32	0,82	3,08
san256	0,71	2,70	2,74	0,81	4,23

Таблиця 4

Результати стиску проблемних та мультимедійних зображень з втратами (абсолютна похибка не більша ніж 2)

1	2	3	4	5	6
problem1	1,24	3,24	3,31	1,39	4,25
problem2	1,24	3,64	3,71	1,40	4,94
problem3	1,23	3,62	3,68	1,40	4,87
problem4	1,24	3,68	3,74	1,38	5,18
problem5	1,24	3,71	3,81	1,33	5,72
problem6	1,24	3,87	3,93	1,39	5,52
problem7	1,24	3,22	3,27	1,39	4,19
problem8	1,23	2,84	2,88	1,40	3,48
problem9	1,25	2,51	2,54	1,40	2,99
problem10	1,23	3,12	3,18	1,40	3,89
boat	1,23	2,84	2,87	1,40	3,72
collie	1,23	3,64	3,71	1,39	5,44
lenna	1,23	3,05	3,08	1,41	3,98
san256	1,23	3,44	3,54	1,35	5,52

Таблиця 5

Результати стиску проблемних та мультимедійних зображень з втратами (абсолютна похибка не більша ніж 4)

1	2	3	4	5	6
problem1	2,37	4,18	4,27	2,45	5,88
problem2	2,39	4,70	4,75	2,44	6,80
problem3	2,37	4,68	4,72	2,44	6,88
problem4	2,45	4,63	4,67	2,36	7,70
problem5	2,29	4,78	4,91	2,21	8,72
problem6	2,31	5,12	5,18	2,30	9,39
problem7	2,36	4,27	4,32	2,42	5,96
problem8	2,38	3,72	3,77	2,48	4,88
problem9	2,40	3,28	3,33	2,54	3,90
problem10	2,37	4,03	4,10	2,49	5,34
boat	2,35	3,71	3,76	2,43	5,45
collie	2,35	4,67	4,70	2,48	6,63
lenna	2,35	3,99	4,02	2,50	5,48
san256	2,35	4,28	4,40	2,40	7,45

З поданих таблицею 2-5 результатів очевидно, що контекстно-базована технологія [1] програє контекстно-базованій технології [2]. Це пояснюється тим, що для полегшення апаратної реалізації та досягнення високої швидкодії в контекстно-базованій технології [1] застосовано спрощену контекстну модель з меншою кількістю контекстів та відмовою від кодування RLE-послідовностей.

Структура алгоритма програми-кодера. Програмну модель безпосередньо реалізовано за допомогою інструменту програмування Borland C++ ver. 5.01. На рис. 2 наведено блок-схему програми-кодера. Програма працює з растровими зображеннями, закодованими у форматі RAW. Коефіцієнт стиску в програмі обчислюється як частка від ділення кількості бітів у вхідному файлі на кількість бітів у вихідному файлі.

На рис.3 наведено блок-схему процедури визначення області, до якої потрапив піксель. Ця функція визначає, до якої із зон належить піксель, і викликає відповідну функцію кодування. Процесу кодування передує перетворення значення інтенсивності. При потраплянні пікселя в центральну зону за точку відліку береться min(p1,p2), в ліву бічну зону - значення, яке передує min(p1,p2), в праву бічну зону - наступне після max(p1,p2) значення.

На рис. 4 наведено блок-схему процедури кодування значення інтенсивності пікселя кодом Голомба-Ріса. Ця функція обирає оптимальне k для кодування значення інтенсивності пікселя кодами Голомба-Риса. Оптимальним для даного контекста вважається k, для якого значення суми довжин кодів оброблених пікселів є мінімальним.

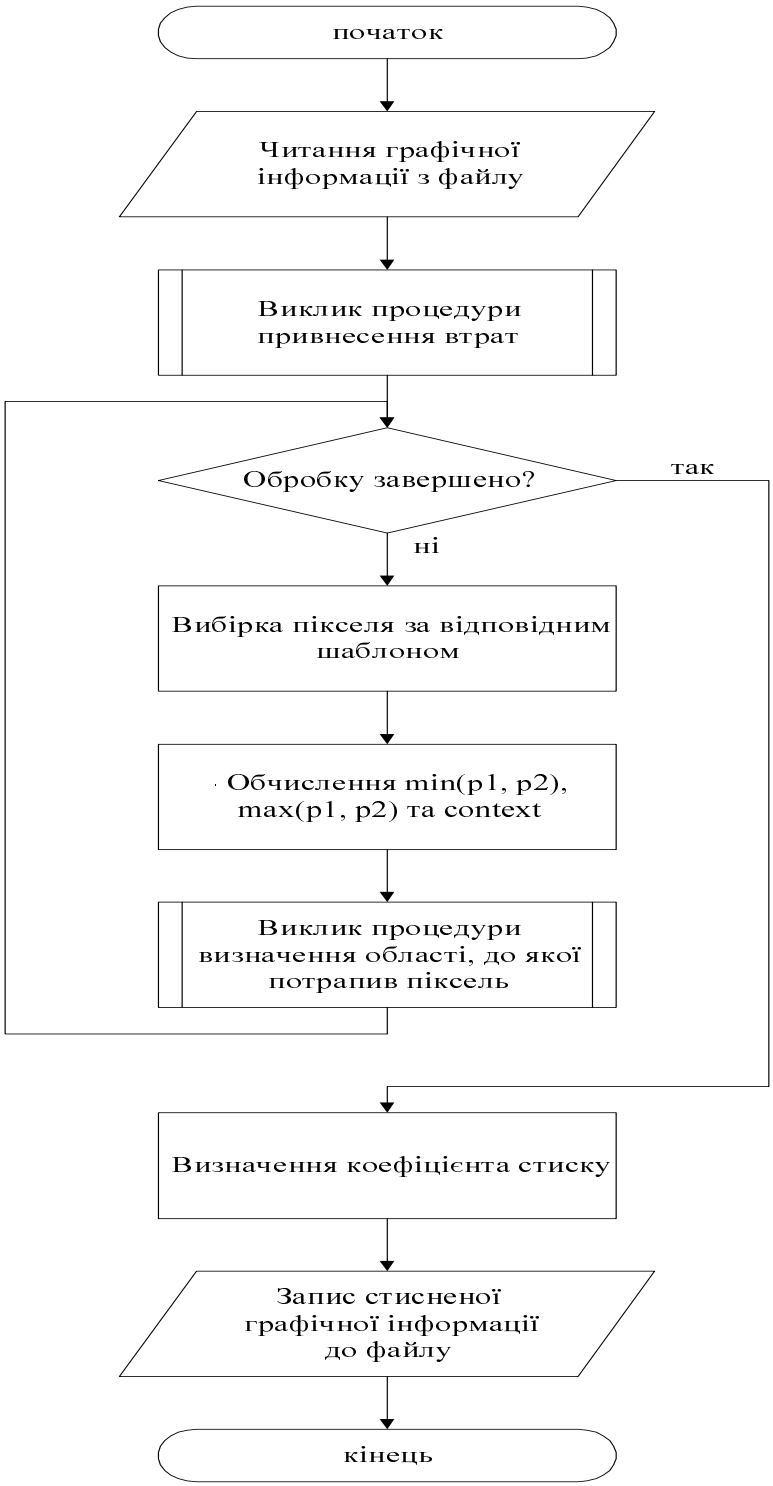

Рис. 2 Блок-схема алгоритму програми-кодера

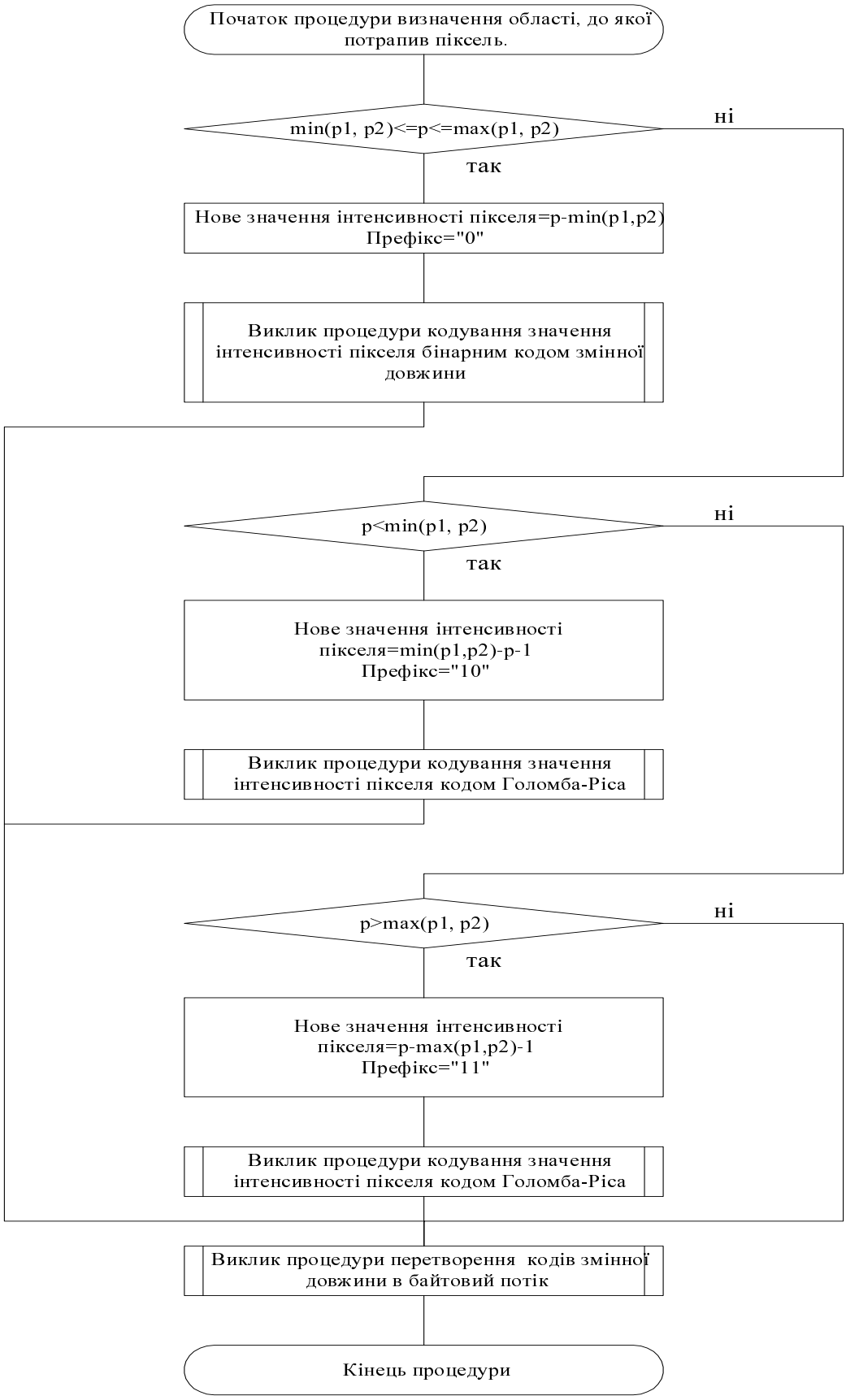

Рис. 3 Блок-схема процедури визначення області, до якої потрапив піксель

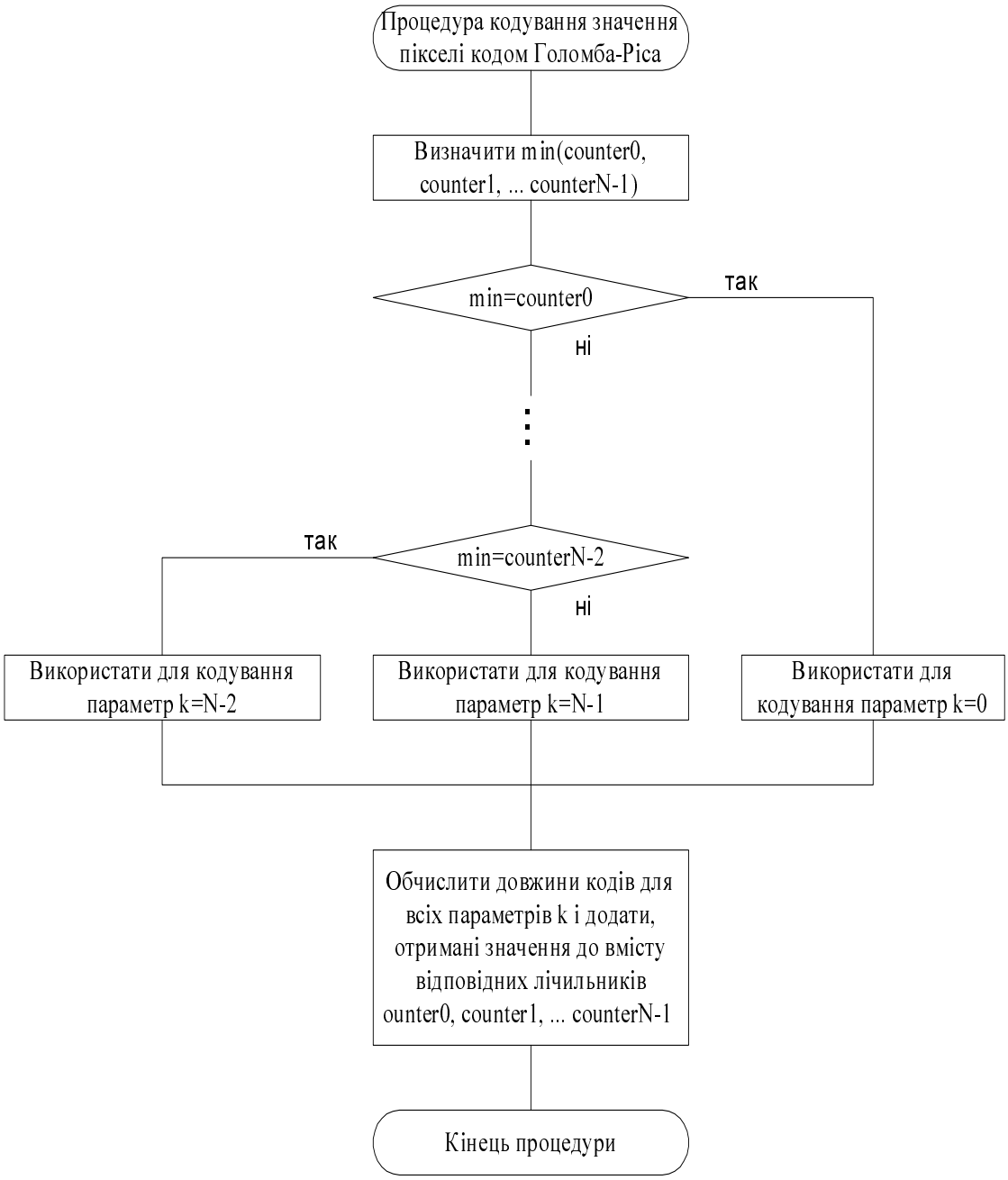

Рис. 4 Блок-схема процедури кодування значення інтенсивності пікселя кодом Голомба-Ріса

Висновки. Запропонований варіант контекстно-базованої технології переважає оригінал за рівнем стиску на 1-3 відсотки і водночас більше відповідає вимогам апаратної реалізації на основі ПЛІС Xilinx.

Отримані результати веріфіковано методом комп'ютерного моделювання.

1. P. G. Howard, J. S. Vitter, "Fast and Efficient Lossless Image Compression," IEEE Computer Society/NASA/CESDIS Data Compression Conference, Snowbird, Utah, March 30--April 1, 1993, pages 351--360. 2. M. J. Weinberger, G. Seroussi, G. Sapiro, "LOCO-I: A Low Complexity, Context-Based, Lossless Image Compression Algorithm," Hewlett-Packard Laboratories, Palo Alto, CA 94304. 3. Справочник по теории вероятностей и математической статистике / Под ред. В.С. Королюка, К., 1978.