

В. С. Глухов, В. М. Хоміць

Національний університет “Львівська політехніка”,  
кафедра електронних обчислювальних машин

## ПІДХІД ДО СТИСНЕННЯ ЗОБРАЖЕНЬ БЕЗ ВТРАТ МЕТОДОМ JPEG-LS

© Глухов В. С., Хоміць В. М., 2017

Розглянуто особливості побудови пристроїв для стиснення монохромних зображень без втрат методом JPEG-LS на сучасних ПЛІС. Детально описано алгоритм стиснення JPEG-LS, його програмну реалізацію мовою С та її часові характеристики.

**Ключові слова:** стиснення без втрат, ПЛІС, JPEG-LS, програмна реалізація, апаратна реалізація

V. Hlukhov, V. Khomits

Lviv Polytechnic National University,  
Computer Engineering Department

## APPROACH TO IMPLEMENTATION OF JPEG-LS LOSSLESS IMAGE COMPRESSION METHOD

© Hlukhov V., Khomits V., 2017

The features of devices for monochrome images lossless compression by JPEG-LS method in modern element base are discussed. Details of compression algorithm JPEG-LS, its C language implementation and its time characteristics are described.

**Key words:** lossless compression, FPGA, JPEG-LS, software and hardware implementation.

### Вступ

У роботі розглянуто особливості виконання стиснення зображень без втрат методом JPEG-LS та його реалізацію у ПЛІС. Методи стиснення без втрат використовуються у сферах, де важливі точність даних та швидкість їх передавання між вузлами системи. Практичне значення має саме його апаратна реалізація. Такі кодери використовують в охоронних системах, системах збирання інформації, супутникових, підводних та інших системах фото- та відеоспостережень, а також в астрономічних інструментах і телескопах. У роботі порівняно алгоритми стиснення зображень та показано переваги застосування JPEG-LS.

### Аналіз літературних джерел

В основу методу стиснення JPEG-LS покладено алгоритм LOCO-I, який описано в роботах [1], [2] та [8]. Детальніша інформація про особливості алгоритму та блок-схема його роботи міститься в інтернет-ресурсах [3], [5], [6]. В роботі [4] описано реалізацію методу стиснення JPEG-LS на ПЛІС фірми Xilinx, а в інтернет-ресурсі [9] міститься його програмна реалізація. Для перетворення форматів відеозображень в ході роботи застосовували конвертор [7]. Формати зображень, які використовують у програмній реалізації, описано в інтернет-ресурсі [10]. Можливу галузь застосування апаратної реалізації описано в [12]. Попередні результати дослідження

наведено в [11]. Питання стиснення зображень нестандартних форматів у згаданих роботах не розглядали.

### Мета роботи

Метою роботи є визначення особливостей алгоритму стиснення зображень нестандартних форматів методом JPEG-LS без втрат інформації, дослідження особливостей його реалізації на ПЛІС, визначення переваг використання цього алгоритму в реалізації вузлів стиснення інформації в бортових системах космічних апаратів.

### Особливості алгоритму JPEG-LS

Метод стиснення JPEG-LS створено на основі алгоритму стиснення LOCO-I. Він істотно відрізняється від методів JPEG та JPEG2000, хоча назва схожа. Цей алгоритм рекомендується використовувати в системах з доволі обмеженими ресурсами, наприклад, космічних станціях чи веб-камерах.

Нижче подано блок-схеми алгоритмів стиснення JPEG (рис. 1), JPEG2000 (рис. 2) та JPEG-LS (рис. 3) для наочного порівняння їхньої структури.



Рис. 1. Схема алгоритму стиснення зображень JPEG: ДКП – дискретне косинусне перетворення; RLE – Run Length Encoding – кодування зі змінною довжиною рядка

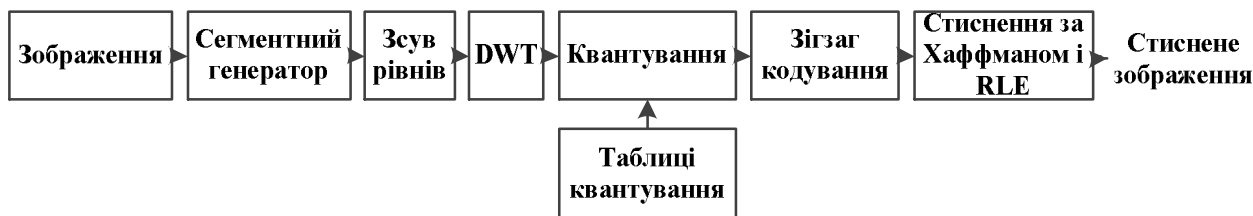


Рис. 2. Схема алгоритму стиснення зображень JPEG2000: DWT – дискретне вейвлет-перетворення

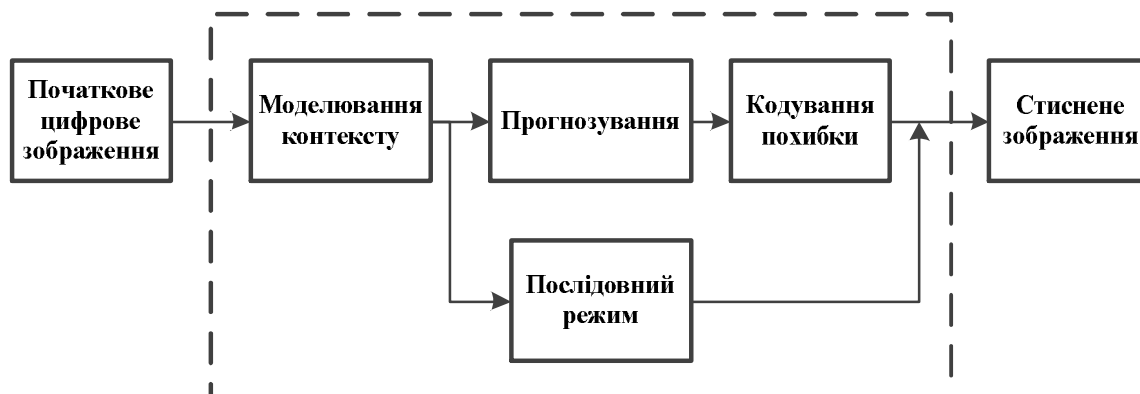


Рис. 3. Схема алгоритму стиснення зображень JPEG-LS

Як можна зауважити, JPEG-LS складається з трьох основних етапів виконання: моделювання, прогнозування та кодування. Основною його відмінністю від інших алгоритмів є модуль прогнозування та робота з контекстом.

Для прогнозування значення чергового пікселя  $x$  використовуються пікселі контексту  $a, b, c, d$  (рис. 4). Залежно від контексту кодер вибирає режим: послідовний (run mode) або регулярний (regular mode) (рис. 5). Послідовний режим вибирають, якщо  $y$  і  $z$ , швидше за все, збігатимуться, регулярний – якщо ні. У разі використання послідовного режиму роботи кодер починає перегляд поточного рядка з пікселя  $x$  і знаходить найбільшу довжину серії пікселів, що збігаються з контекстним пікселем  $a$ . Отже, в межах поточного рядка отримують послідовність однакових пікселів, які збігаються за значенням з відомим пікселем  $a$ . Після цього кодується довжина послідовності. У разі використання регулярного режиму для обчислення прогнозованого значення пікселя  $x$  ( $P_x$ ) використовують значення пікселів  $a, b$  і  $c$ . Також обчислюється так звана помилка прогнозу (Errval). Її значення дорівнює різниці значення  $x$  і  $P_x$ . Errval коригується, а потім кодується за допомогою кодів Голомба. Код Голомба [8] залежить від  $a, b, c, d$  і Errval цих самих пікселів.

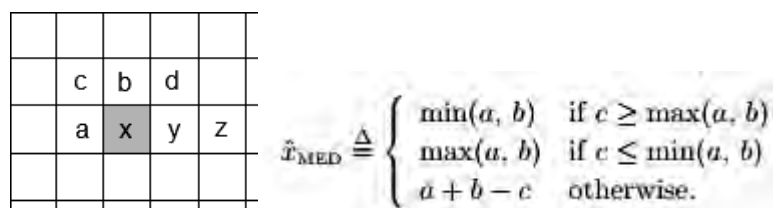


Рис. 4. Контекстні пікселі для пікселя  $x$  та розрахунок його прогнозованого значення

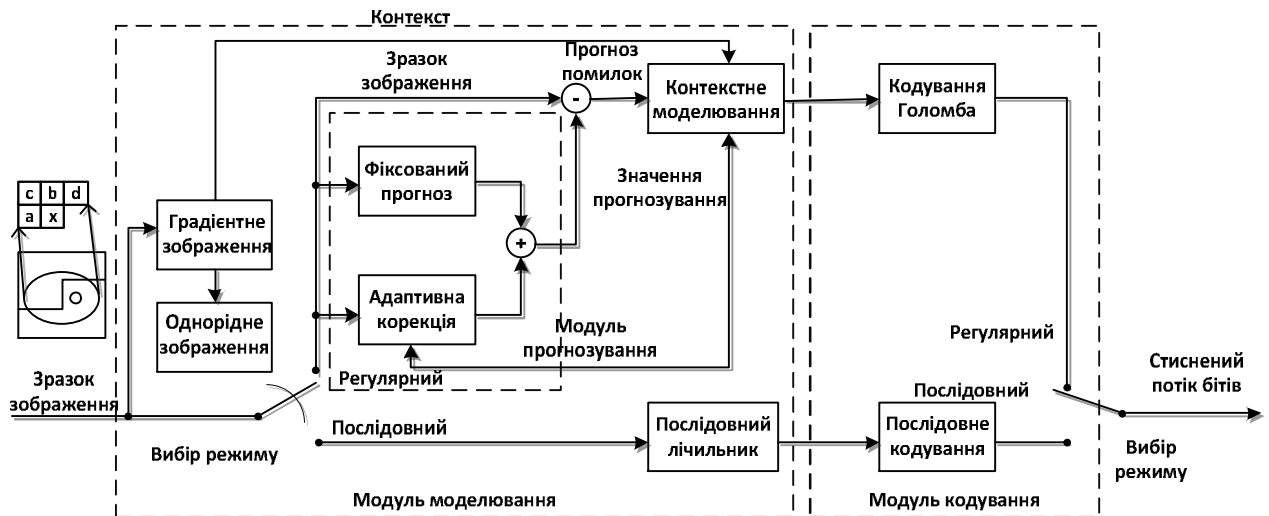


Рис. 5. Схема алгоритму стиснення зображень JPEG-LS

Прогнозоване значення для поточної вибірки  $x$  повинно залежати від  $a, b, c$  і  $D$ . У Loco-I / JPEG-LS, вузол передбачення виконує примітивний тест для виявлення вертикального або горизонтального краю.

Вузол передбачення перемикається між трьома простими прогнозами: він вибирає  $b$  у тих випадках, коли вертикальне ребро існує зліва від поточного місця розташування,  $a$  – в разі горизонтального краю над поточним місцезнаходженням, або  $a+b-c$ , якщо жодне ребро не виявлено. Прогнозоване значення визначається як медіана трьох фіксованих прогнозів:  $a, b, a+b-c$ .

Програму jpeg-ls v2.2 [9] розроблено в лабораторії Hewlett-Packard в 1995 р., вона є у вільному доступі для дослідницьких чи навчальних цілей. Ця програма виконує стискання та

розтискання зображення методом jpeg-ls, а також надає можливість виконати стиснення з втратами. Алгоритм роботи програми наведено на рис. 6. Програма працює з трьома форматами зображень – bmp, png, ppm [10]. За допомогою ресурсу [7] у ході роботи створено тестові зображення для дослідження роботи програми.

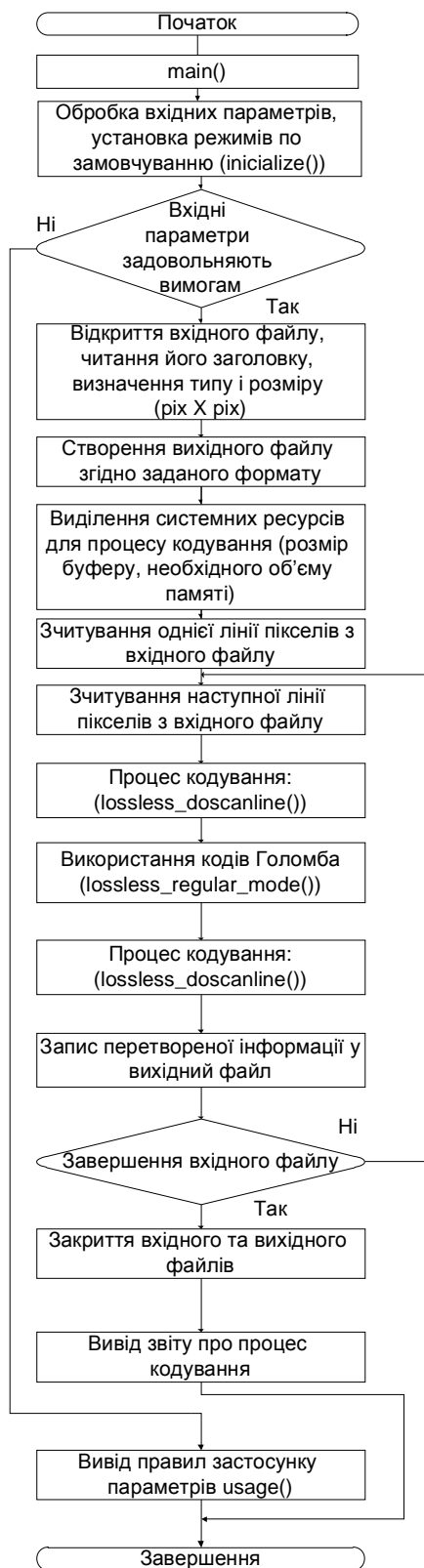


Рис. 6. Схема алгоритму роботи програми

Паке́т програми складається з файла JLSEncoder.exe, текстового документа README та двох папок – Encoder, Decoder з початковими кодами С. В описі програми зазначено, що коди С не є завершеною програмою і лабораторія НР не гарантує їх коректної роботи. Також додано пояснення до роботи програми та вказано команди для командного рядка в консолі.

Для визначення можливостей програми jpeg-ls v2.2 використано вже готовий додаток JLSEncoder.exe. Спочатку було стиснено шість зображень форматів bmp, pgm, ppm з розмірами сторін до 2000 пікселів. Усі стиснення здійснено в режимі “без втрат”. Як зразки вибрано три монохромні зображення формату pgm та три кольорові формати ppm і bmp. Результати тестування містить табл. 1.

Як видно з табл. 1, досліджувана програмна реалізація методу jpeg-ls найкраще стискає bmp-зображення та зображення з малою кількістю відтінків. Максимальна кількість відтінків, що може обробити ця програма, – 256.

Співвідношення сторін зображення (табл. 1) не впливає істотно на результати стиснення. На рис. 6 та 7 наведено результати стиснення зображень більших розмірів, де видно співвідношення між кількістю рядків пікселів у зображенні та часом виконання стиснення. Для зображень формату pgm за замовчуванням вибирають лінійний режим, для зображень ppm та bmp краще працює регулярний режим. У цьому дослідженні всі перетворення здійснювались у лінійному режимі для однорідності результату.

Подальший аналіз здійснювався після редагування та компіляції відкритих С-кодів програми jpeg-ls v2.2 та лише з зображеннями формату pgm, тобто з монохромними. За результатами (табл. 1, 2) стиснення зображень різних розмірів (cols – кількість пікселів у рядку, rows – кількість рядків у зображенні) визначено середній час стиснення одного рядка пікселів, одного пікселя у зображенні, коефіцієнт стиснення та найоптимальніший розмір зображення для ефективного стиснення досліджуваною програмою. Надалі ці розміри будуть використовуватись для розроблення вузла стиснення зображень на базі ПЛІС.

Таблиця 1

#### Порівняння результатів стиснення програмою jpeg-ls v2.2

Ім'я та формат файла	К-сть пікселів по горизонталі	К-сть пікселів по вертикалі	Час стиснення, с	Коефіцієнт стиснення
FNS1694a.pgm	200	1800	0.021	3.70 : 1
tatoo.pgm	200	2800	0.107	1.93 : 1
Forest1.pgm	1280	960	0.570	1.76 : 1
forest2.ppm	1280	960	0.185	1.68 : 1
train.bmp	1471	870	–	7.74 : 1
wolf.bmp	425	396	–	6.00 : 1

Унаслідок стиснення зображень, які відповідають завданню сканування земної поверхні з борта космічного апарата (ширина рядка – 18 000 пікселів), отримано результати, які містить рис. 7.

Для заданої ширини зображення (18000 пкс), якщо зростає кількість рядків, середній час кодування одного рядка зменшується до моменту, коли зображення набере форми квадрата (у цьому випадку до розміру зображення 18000×18000 пкс). З подальшим зростанням кількості рядків середній час кодування рядка практично не змінюється, швидкодія також не зростає, лише збільшується коефіцієнт стиснення. Тобто оптимальним для кодування за швидкістю є зображення, в якому кількість рядків не менша за ширину зображення (в пкс).

На графіках також показано лінію тренду. Найкраще для оцінки підходить степенева функція. Оцінювальне значення часу стиснення зображення для заданої ширини рядка (18000 пкс):  $y=0,004x^{-0.169}$  (x – кількість рядків).

Для кращого вивчення властивостей стиснення пікселів проведено додаткове дослідження, у ході якого стискалися квадратні зображення формату 4x4, 8x8, 16x16, ... 16384x16384 пкс (табл. 3, сторона кожного квадрата дорівнює степені 2).

Як видно (рис. 8), доцільніше стискати великі квадратні зображення розміром від 256x256 пкс і більші. Зі зростанням розмірів квадрата до 256x256 пкс час кодування одного пікселя різко зменшується (рис. 8), але чим більше зображення, тим менш помітна різниця у часі, хоча вона і є. Лінія тренду на графіку описує зміну часу кодування (найоптимальніше підходить степенева функція), описується як  $y=0,0003x^{0.7357}$ .

Визначено швидкодію програми: файл розміром 1582032 Кб, або 1,582032 Гб, кодується 68,644 с. Швидкість кодування в Гб/с дорівнює  $1,582032/68.644 \approx 23$  Мб/с (мегабіт за секунду), що відповідає приблизному часу кодування одного 8-бітного пікселя 340 нс.

Усі дослідження виконано на комп'ютері з двоядерним процесором Intel (R) Pentium (R), CPU G2020 з тактовою частотою 2,90 ГГц, оперативною пам'яттю місткістю 2 Гб та 32-розрядною операційною системою Windows 7.

Таблиця 2

**Результати стиснення за заданої ширини зображення – 18000 пікселів**

Розмір зображення (cols x rows)	Час кодування, с	Коефіцієнт стиску	Швидкодія, пікселів/с	Середній час кодування одного рядка, мс	Середній час кодування одного пікселя, мкс
18000x1000	4,010	(6.339:1)	13151	4,01	0,223
18000x2000	7,301	(6.918:1)	14446	3,65	0,203
18000x3000	10,343	(7.265:1)	15296	3,45	0,192
18000x4000	12,917	(7.525:1)	16330	3,23	0,179
18000x5000	15,226	(7.729:1)	17317	3,05	0,169
18000x6000	17,971	(7.893:1)	17606	3,00	0,166
18000x7000	20,405	(8.030:1)	18091	2,92	0,162
18000x8000	23,103	(8.148:1)	18261	2,89	0,160
18000x9000	24,741	(8.251:1)	19183	2,75	0,153
18000x10000	27,129	(8.341:1)	19438	2,71	0,151
18000x11000	29,703	(8.421:1)	19529	2,70	0,150
18000x12000	32,073	(8.493:1)	19730	2,67	0,148
18000x13000	34,180	(8.557:1)	20057	2,63	0,146
18000x14000	34,930	(8.614:1)	21099	2,50	0,139
18000x15000	37,020	(8.667:1)	21367	2,47	0,137
18000x16000	39,017	(8.714:1)	21625	2,44	0,135
18000x17000	41,154	(8.757:1)	21784	2,42	0,134
18000x18000	42,901	(8.797:1)	22126	2,38	0,132
18000x19000	45,179	(8.833:1)	22177	2,38	0,132
18000x20000	47,207	(8.866:1)	22342	2,36	0,131
18000x21000	49,560	(8.896:1)	22463	2,36	0,131
18000x22000	51,778	(8.926:1)	22406	2,35	0,131
18000x23000	54,383	(8.953:1)	22303	2,36	0,131
18000x24000	57,207	(8.978:1)	22124	2,38	0,132
18000x25000	59,344	(9.001:1)	22216	2,37	0,132
18000x26000	61,777	(9.023:1)	22194	2,38	0,132
18000x27000	64,507	(9.043:1)	22072	2,39	0,133
18000x28000	66,801	(9.063:1)	22104	2,39	0,133
18000x29000	69,515	(9.081:1)	22000	2,40	0,133
18000x30000	70,405	(9.088:1)	22470	2,35	0,130

**Реалізація методу JPEG-LS на ПЛІС**

Завдяки швидкості, простоті та адаптивності алгоритм стиснення JPEG-LS можна використовувати не тільки в програмних, але і в апаратних реалізаціях. Однією з успішних реалізацій є ядро ПЛІС виробництва фірми Alma Technologies. Нижче наведено структуру (рис. 9) та опис кожного компонента реалізації.

Модуль аналізатора потоку (Stream Parser) визначає параметри стиснення, готує дані для стиснення, перевіряє їх на відсутність помилок.

Пам'ять контекстних змінних (Context Variables Memory) зберігає поточні значення параметрів, які використовує і оновлює модуль контекстного моделювання.

Пам'ять конфігурації (Configuration Memory) зберігає параметри, які визначають режим роботи пристрою.

Модуль регістрів стану/контролю (Control-status Registers) містить чотири 16-бітні регістри стану і регістрів контролю.

Вхідний буфер (Input Buffer) складається з модуля пам'яті та двох комірок для пам'яті операцій читання і запису. Цей компонент зберігає вхідні дані вибірки, які подаються на ядро інтерфейсу.

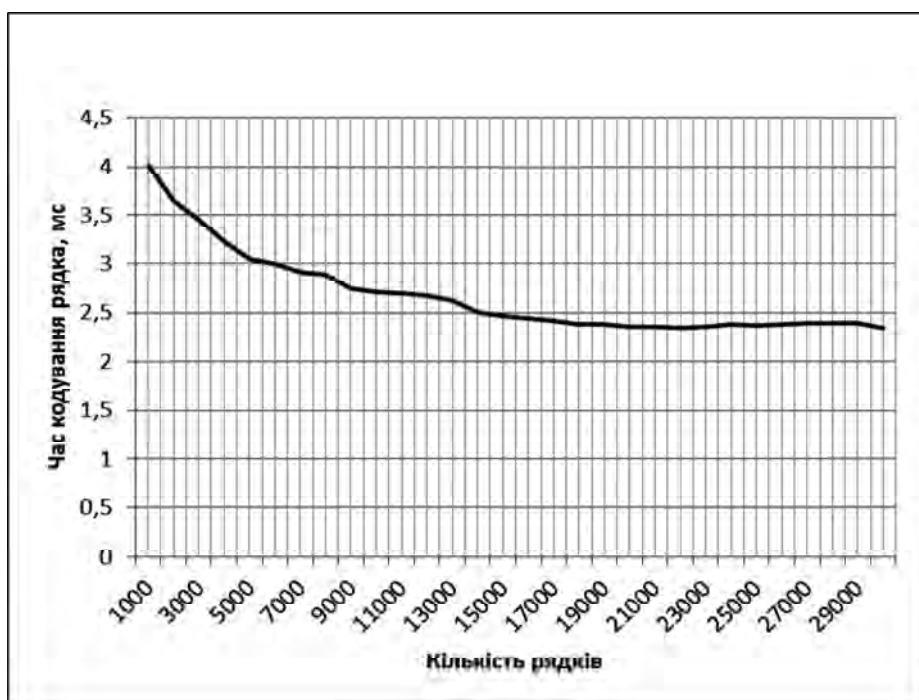


Рис. 7. Стівідношення середнього часу кодування одного рядка зображення та кількості рядків за постійної ширини рядка – 18000 пкс

Таблиця 3

### Результати стиснення зображень типу квадрат

Файл (cols x rows)	Час кодування, с	Коефіцієнт стиску	Швидкодія, Ksymbol/sec	Середній час кодування одного рядка, мс	Середній час кодування одного пікселя, мкс
4x4	0.006	(0.462:1)	8	1,50	375,000
8x8	0.004	(0.835:1)	19	0,50	62,500
16x16	0.004	(1.286:1)	187	0,25	15,625
32x32	0.004	(1.743:1)	750	0,13	3,906
64x64	0.006	(2.123:1)	2000	0,09	1,465
128x128	0.009	(2.314:1)	5333	0,07	0,549
256x256	0.023	(2.258:1)	8348	0,09	0,351
512x512	0.084	(2.017:1)	9143	0,16	0,320
1024x1024	0.319	(2.287:1)	9630	0,31	0,304
2048x2048	1.160	(3.666:1)	10593	0,57	0,277
4096x4096	4.056	(5.524:1)	12118	0,99	0,242
8192x8192	13.121	(6.780:1)	14984	1,60	0,196
16384x16384	44,161	(12.022:1)	17808	2,70	0,165

Модуль контекстного моделювання (Context Modeler) здійснює операції обчислення помилки прогнозування. Він звертається до буфера рядка і читає чотири контекстні значення, необхідні для визначення контексту. Крім того, цей модуль виконує процедури серединного прогнозу, виявлення пошуку краю зображення і всі обчислення етапу прогнозування помилки. Нарешті, цей модуль оновлює змінні контекстної пам'яті новими значеннями статистичних параметрів вибраного контексту.

Модуль кодування Голомба (Golomb Coder), за помилкою передбачення і параметрами стиснення, формує код Голомба, готує його для збереження у внутрішній пам'яті FIFO.

Модуль перевірки синтаксису потоку (Stream syntaxer) перевіряє сформований потік даних на наявність помилок.

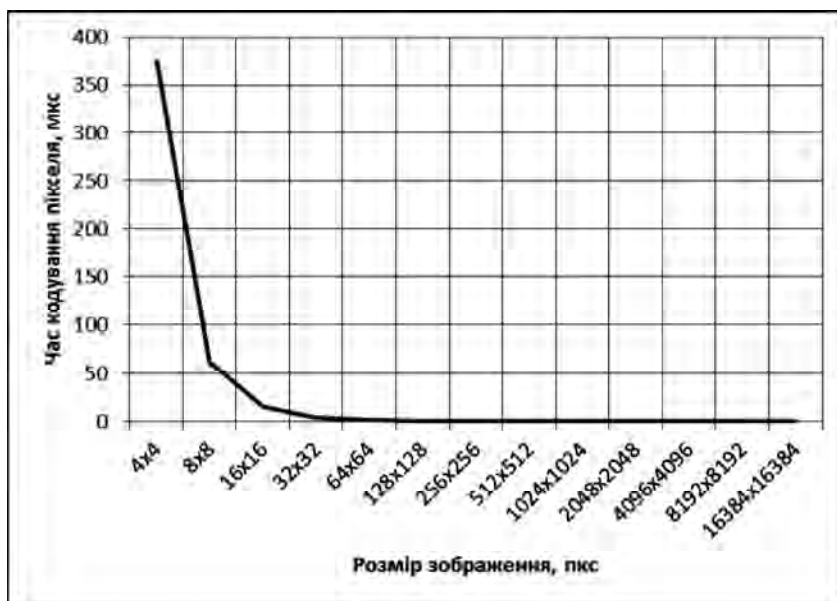


Рис. 8. Співвідношення часу кодування одного пікселя та розміру зображення

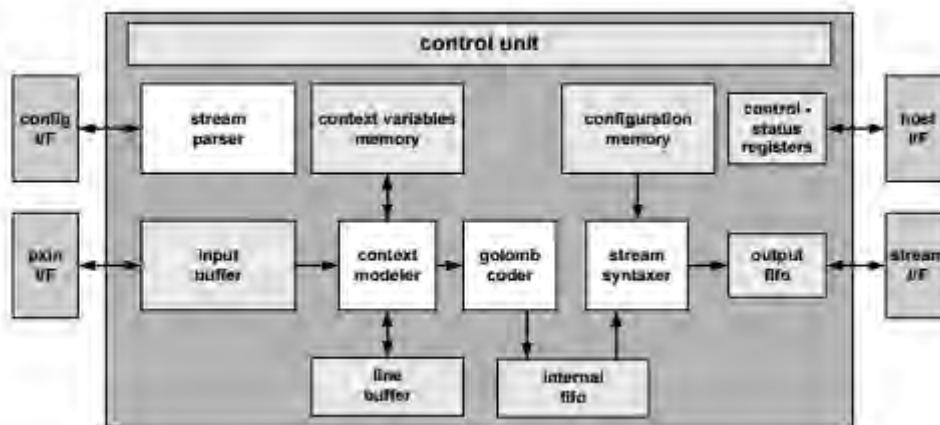


Рис. 9. Структура ядра ПЛІС фі. Alma Technologies

Буфер рядків (Line Buffer) зберігає попередньо закодований стиснутий рядок та кожен компонент зображення.

FIFO зберігає кодові слова, які формуються у модулі кодування Голомба, і виводить їх на модуль перевірки синтаксису потоку.

Описану реалізацію можна взяти за основу створення апаратних реалізацій вузлів стиснення спеціального призначення.



## Висновки

У роботі досліджено особливості алгоритму стиснення зображень без втрат JPEG-LS, який описано мовою С, а також особливості його реалізації на ПЛІС, визначено найкращі формати зображення для програмного стиснення, а також часові характеристики алгоритму за його програмної реалізації для форматів зображення bmp, pgm, ppm, оцінювальне значення продуктивності програмної реалізації – 0,34 мкс/пкс. Описану в роботі апаратну реалізацію можна взяти за основу створення на ПЛІС вузлів стиснення зображень нестандартних форматів для космічних апаратів.

1. Vajnberger M., Seroussi G., Schapiro G. *The Loco-I stysnennja zobrazhen bez vtrat. Alorytm: pryntsyty i standartyzatsiji v JPEG-LS // Hewlett-Packard Laboratories Technical Report No. HPL-98-193R1, lystopad 1998, pereroblene zhovtnja +1999. IEEE Trans. Obrobka zobrazhen, Vol. 9 serpnja 2000 roku, pp. 1309–1324* 2. Vajnberger M., Seroussi G., Schapiro G. *The Loco-I: nyzka skladnist, zasnovana na konteksti, stysnennja zobrazhen bez vtrat Alorytm // Proc. Konferentsija IEEE Data Compression, Snowbird, shtat Juta, berezen-kviten 1996 roku.* 3. *DISKRETNAYA MATEMATIKA: ALGORITMY. JPEG, JPEG 2000, JPEG-LS. Szhatie izobrazhenij s poterjami i bez [Elektronnyj resurs] : [Veb-sajt]. – Elektronni dani. – Rezhym dostupu: <http://rain.ifmo.ru/cat/view.php/theory/data-compression/jpeg-2006>.* 4. *JPEG-LS Encoder Core (Numerically and Near Lossless Compression) Design Specification. 2007 – 2013 ALMA Technologies.* 5. *This page describes the description of a JPEG-LS encoder implemented by the author in VHDL for use in projects on the FPGA. [Electronic resource]. – Access mode: <http://jpegls.narod.ru>* 6. Brailovskij I. V. *Efektivnoe szhatie izobrazhenij s ispolzovaniem obobschennyh intervalnyh preobrazovanij [Elektronnyj resurs] – Rezhym dostupu: [http://www.ict.edu.ru/ft/002147/sb4\\_page86\\_95.pdf](http://www.ict.edu.ru/ft/002147/sb4_page86_95.pdf).* 7. *Konverter fajlov. Onlajn i Besplatno. [Elektronnyj resurs] – Rezhym dostupu: <https://convertio.com>.* 8. *Vikiznanie. Kod Golomba [Elektronnyj resurs] : [Veb-sajt]. – Elektronni dani. – Rezhym dostupu: [http://www.wikiznanie.ru/wikipedia/index.php/%D0%9A%D0%BE%D0%B4\\_%D0%93%D0%BE%D0%BB%D0%BE%D0%BC%D0%B1%D0%B0](http://www.wikiznanie.ru/wikipedia/index.php/%D0%9A%D0%BE%D0%B4_%D0%93%D0%BE%D0%BB%D0%BE%D0%BC%D0%B1%D0%B0).* 9. *Laboratoria Hewlett-Packard [Elektronnyj resurs] : [Veb-sajt]. – Elektronni dani. – Rezhym dostupu: [http://www.labs.hp.com/research/info\\_theory/loco/locodown.htm](http://www.labs.hp.com/research/info_theory/loco/locodown.htm).* 10. *Vikipedija. Format Netpbm [Elektronnyj resurs] : [Veb-sajt]. – Elektronni dani. – Rezhym dostupu: [https://en.wikipedia.org/wiki/Netpbm\\_format](https://en.wikipedia.org/wiki/Netpbm_format).* 11. *Gluhov V. S., Homits V. M. Stysnennja zobrazhen bez vtrat metodom jpeg-ls na PLIS – Tretij naukovyj seminar “Kiberfizychni sustemy: dosjagnennja ta vyklyky”. PROGRAMA. Natsionalnyj universytet “Lvivska politehnika”, 13–14 chervnja 2017 r. Lviv, Ukraina. – 26–37 p.* 12. *Hlukhov Valerii, Lukenyuk Adolf, Shenderuk Sergii. Satellite scientific data collection and accumulation system as a basis for cyber-physical systems construction. Advances in Cyber-Physical Systems. Volume 1. Number 2. Lviv Polytechnic National University. 2016. – P. 77–86.*