

Я. М. Крайник, аспірант,
codebreaker7@ukr.net

О. О. Денисов, здобувач
maildenisov@gmail.com

Чорноморський державний університет імені Петра Могили
м. Миколаїв, вул. 68 Десантників, 10

КОНВЕЄРНА АРХІТЕКТУРА LDPC-ДЕКОДУВАННЯ НА БАЗІ ПЛІС З ВИКОРИСТАННЯМ МОДИФІКОВАНОГО АЛГОРИТМУ МІНІМАЛЬНОЇ СУМИ

У статті представлено розроблений метод побудови конвеєрної архітектури частково паралельного LDPC-декодера (Low Density Parity Check – укр. низька щільність перевірки на парність) на базі ПЛІС (програмована логічна інтегральна схема). Побудова архітектури передбачає попередню обробку матриці перевірки парності з метою уникнення колізій доступу. Запропонована архітектура реалізує модифікований алгоритм декодування мінімальної суми, який дає змогу зменшити кількість необхідних ітерацій та, відповідно, збільшити пропускну здатність. Запропоновані положення дозволяють суттєво підвищити пропускну здатність декодера.

Ключові слова: LDPC-декодер, матриця перевірки парності, алгоритм мінімальної суми.

Вступ. Нерегулярні LDPC-коди [1] (Low Density Parity Check – укр. низька щільність перевірки на парність) з довільним розташуванням значущих елементів здатні забезпечувати показники виправлення помилок, які є найближчими до границі Шеннона серед усіх відомих типів кодів [2]. Побудова апаратних декодерів для таких кодів передбачає значну складність з точки зору забезпечення високої пропускну здатності. Ця обставина пояснюється довільним розташуванням значущих елементів матриці перевірки парності (МПП), що накладає жорсткі обмеження на можливість спрощення архітектури декодера та організації паралельних обчислень.

Конвеєрна організація декодера дозволяє значно підвищити швидкодію і широко використовується для спеціальних типів МПП. Проте застосування конвеєра для довільної матриці не є можливим, оскільки не гарантує уникнення колізій при обробці вхідних даних. Виникнення колізій означає необхідність заміни паралельних операцій послідовними (декодери такого типу МПП найчастіше відносяться до частково паралельних). МПП має пройти попередню обробку для забезпечення коректності результатів та уникнення колізій при декодуванні. Це дозволить застосувати конвеєрну організацію обчислень та збільшити пропускну здатність.

Аналіз останніх джерел досліджень і публікацій. Конвеєрна архітектура LDPC-

декодера представлена в багатьох дослідженнях, зокрема, у [3, 4]. Організація обчислювального конвеєра означає паралельне виконання основних етапів декодування над різними даними, що значно підвищує пропускну здатність [4] (вдається досягти результатів у кілька сотень Мбіт/с та, навіть, Гбіт/с). Незважаючи на це, застосування конвеєрної організації у в зазначених роботах проведено лише для таких МПП, для яких конвеєрна обробка не викликає колізій. Тому необхідно дослідити можливість застосування конвеєрної архітектури для довільних МПП.

У роботі [5] проводиться дослідження архітектур LDPC-декодерів. Зокрема, у цій роботі надаються рекомендації щодо розробки декодерів для довільних МПП, серед яких вказується необхідність попередньої обробки МПП, акцентується увага на проблемах, що необхідно вирішувати при попередній обробці МПП. Проте, як саме необхідно проводити цю обробку, в роботі не зазначається.

У роботах [6, 7] проводиться дослідження побудови декодера для довільних МПП. Описані особливості уникнення колізій та організації паралельних обчислень з урахуванням обмежень. Однак швидкодія отриманих рішень, незважаючи на те, що вона зіставна з відомими аналогами, може бути збільшена.

Завдяки простоті реалізації, великим можливостям для розпаралелювання операцій

та достатній якості декодування алгоритм мінімальної суми [8] найчастіше використовується при апаратній реалізації декодера. Однак цей алгоритм є чутливим до таких параметрів декодера, як МПП, розрядність вхідних даних, представлення даних та операцій над ними, внаслідок чого виникають також модифіковані версії цього алгоритму [9]. Залежно від зазначених параметрів змінюється швидкість декодування за ітераціями. Тому важливим аспектом дослідження є пошук варіантів модифікації алгоритму, які дозволять зменшити кількість ітерацій, що необхідні для декодування.

Формулювання мети дослідження.

Метою роботи є підвищення швидкодії декодера нерегулярних LDPC-кодів на основі довільних МПП за рахунок розробки методу побудови конвеєрної архітектури частково паралельного декодера та модифікації алгоритму декодування мінімальної суми.

Основна частина. Реалізація декодера за методом, що пропонується, передбачає виконання таких трьох етапів:

- 1) попередня обробка МПП;
- 2) організація конвеєрної архітектури декодера;
- 3) реалізація модифікованого алгоритму мінімальної суми.

Введемо наступні позначення для важливих характеристик МПП, відповідного їй графу Таннера та для опису обчислень алгоритму мінімальної суми у процесі декодування. МПП містить M рядків та N стовпців. При цьому, максимальна кількість значущих елементів у окремому рядку – w_{rmax} . Граф Таннера містить вузли перевірки $c_i, i = 1..M$ та вузли значень $v_j, j = 1..N$. Алгоритм декодування мінімальної суми є ітеративним алгоритмом на основі обміну повідомленнями між вузлами графу Таннера. Вузли перевірок обчислюють значення повідомлення α_{ij} для кожного вузла значення на основі отриманих від з'єднаних вузлів значень повідомлень β_{ij} , та навпаки. Остаточні значення за результатами ітерації декодування зберігаються в пам'яті Z та визначаються як

$$z_j = z_{jstart} + \sum_{i \in I} \alpha_{ij}, \quad (1)$$

де z_{jstart} – початкове значення для j -го вузла; I – множина всіх вузлів перевірки, що підключені до j -го вузла значень. При цьому з урахуванням того, що у ході ітерації остаточне значення z_j визначається не одразу, то використовується також пам'ять проміжних результатів ітерації Z_{temp} з елементами z_{tempj} .

Попередня обробка МПП проходить у декілька кроків. На першому кроці використовується операція перестановки стовпців матриці з метою розділення на інтервали, які містять не більше одного значущого елемента. На цій стадії визначається ширина кожного інтервалу $width_i$ та їх кількість n . У процесі заповнення інтервалів необхідно відстежувати, щоб стовпець, який додається add_col , не був доданий до інтервалу, що вже містить такий стовпець $check_col$, який має значущий елемент з однаковим індексом рядка:

$$\begin{aligned} & \forall row_ind(add_col, j_1) \\ & \neg \exists row_ind(add_col, j_1) = \\ & row_ind(check_col, j_2), \\ & j_1 = 1..N_1, j_2 = 1..N_2, \end{aligned} \quad (2)$$

де row_ind – функція, що повертає індекс рядка; N_1, N_2 – кількість значущих елементів у відповідних стовпцях. Якщо ця умова виконується, то стовпець може бути доданий до інтервалу, інакше необхідно спробувати додати його до наступного інтервалу. Блок-схему описаного алгоритму перестановки стовпців матриці зображено на рис. 1.

У тому випадку, коли наявні такі стовпці, що їх не можна додати до будь-якого інтервалу, слід збільшити кількість інтервалів. У результаті цього кроку отримується масив відповідності для початкової та результуючої матриці.

На другому кроці при обробці матриці використовується інша операція елементарного перетворення матриці – перестановка рядків. На цьому етапі забезпечується виконання умови відсутності однакових індексів у m попередніх та послідовних рядках відносно поточного. Значення m визначається характеристиками конвеєра обробки.

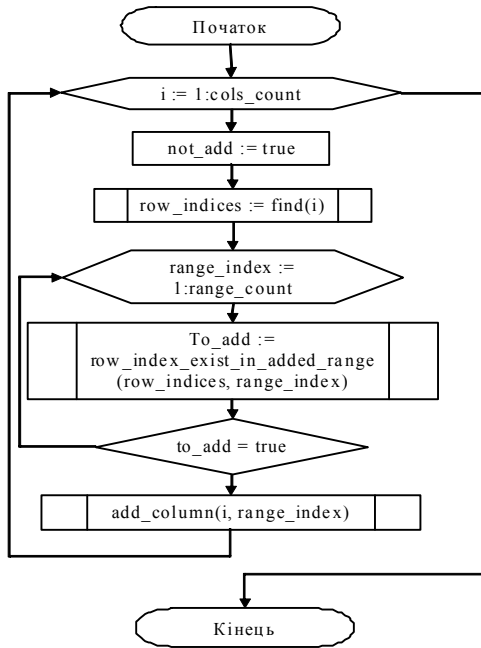


Рис. 1. Блок-схема алгоритму перестановки стовпців для попередньої обробки матриці

Конвеерна архітектура декодера залежить від обраного алгоритму декодування. Організація конвеєра передбачає, що вихід одного обчислювального блоку є входом іншого. Відповідно до цього та з урахуванням обчислювальних операцій, що використовуються в алгоритмі мінімальної суми, пропонується наступна організація конвеєра (рис. 2).

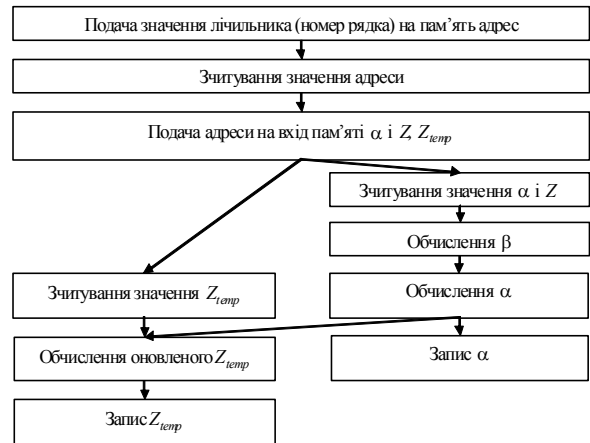


Рис. 2. Структурна організація конвеєра обчислень для декодера

Ця організація конвеєра при повному завантаженні всіх складових на одному циклі забезпечує обробку одного рядка з частковою обробкою стовпців значущих елементів у рядку. Таким чином, загальний час обробки даних на основі МПП з M рядками становитиме

$$time_total = M + lag_{pipeline}, \quad (3)$$

де $lag_{pipeline}$ – загальна затримка конвеєра, що визначається кількістю стадій обробки, а також залежить від характеристик МПП.

Загальну діаграму зміни частини сигналів при конвеєрній обробці зображено на рис. 3.

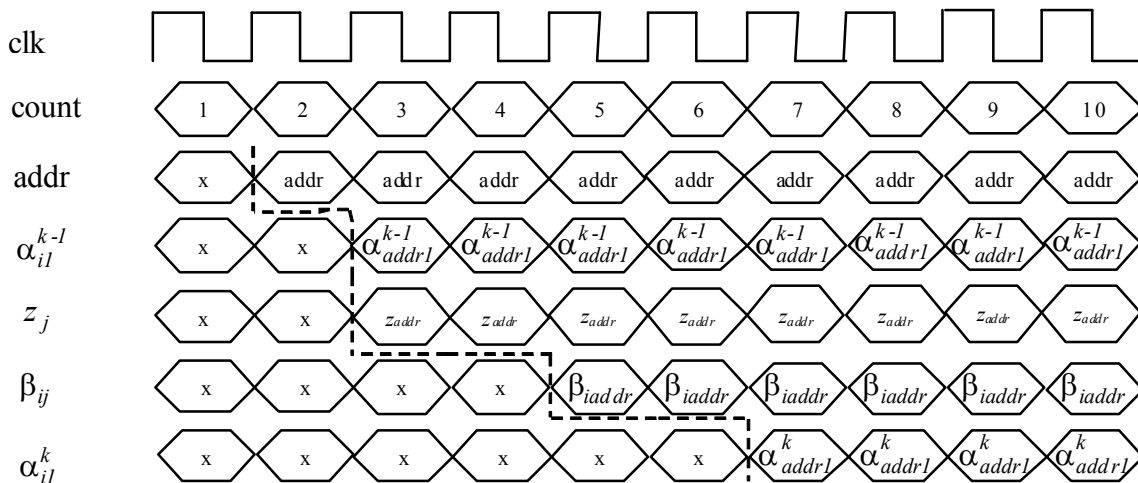


Рис. 3. Загальна діаграма роботи конвеєру декодування

Частина значень сигналів на діаграмі позначена неактивною через те, що конвеєр на початку роботи завантажений необхідними даними не на всіх стадіях, а лише на частині.

Через це наявна затримка для різних сигналів, яка виділена на рисунку.

Алгоритм мінімальної суми передбачає те, що на початок кожної ітерації у Z та

Z_{temp} знаходяться однакові значення. Проте у тому випадку, коли у вхідному повідомленні наявна відносно невелика кількість помилок, то бажаною характеристикою алгоритму декодування є здатність виправляти якомога більше некоректних значень за одну ітерацію, не вносячи при цьому помилки в коректні позиції. Це дозволяє зменшити загальну кількість ітерацій, а відповідно, підвищити швидкодню. Такого результату вдається досягти в тому випадку, якщо не проводити запис з пам'яті проміжних значень Z_{temp} у пам'ять Z , а використовувати Z_{temp} як Z на наступній ітерації. При цьому, значення в іншій пам'яті не змінюються. Фактично, це відповідає обміну значеннями між Z та Z_{temp} .

$$\begin{aligned} buf &= Z; \\ Z &= Z_{temp}; \\ Z_{temp} &= buf. \end{aligned} \tag{4}$$

Таким чином, значення у Z та Z_{temp} на початку ітерації для модифікованого алгоритму не є однаковими.

У випадку реалізації для ПЛІС це може бути реалізовано за допомогою простого селектора пам'яті, що змінює своє значення при зміні номеру ітерації:

$$sel_val = num_iter \bmod 2, \tag{5}$$

де num_iter – номер поточної ітерації. Представлення алгоритму з точки зору графу Таннера зображено на рис. 4.

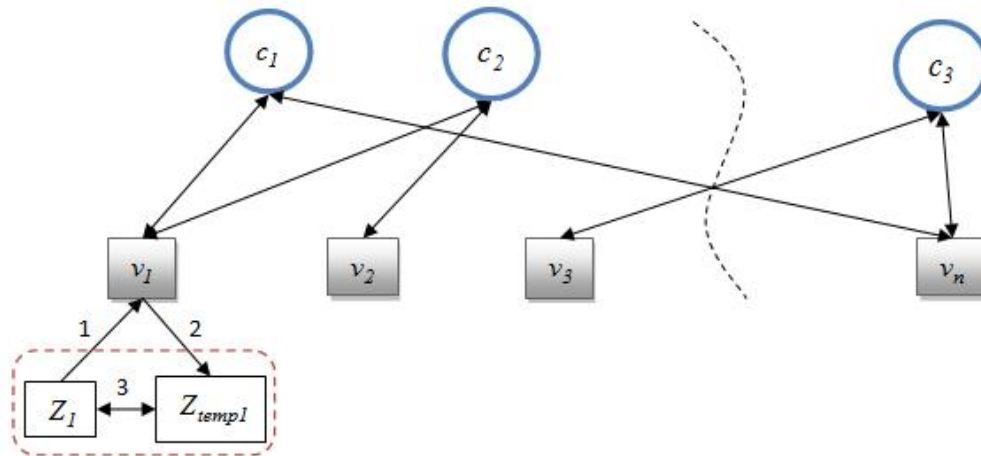


Рис. 4. Послідовність виконання операцій з пам'яттю за модифікованим алгоритмом

Розроблені положення реалізовані для декодера МПП, що містить 16000 стовпців та 4000 рядків. Реалізація виконана за допомогою мови схемотехнічного опису VHDL. Для симуляції розробленого рішення використо-

увалось програмне середовище Mentor-Graphics ModelSim Starter Edition 10.1. На рис. 5 зображено скріншот початку процесу декодування.

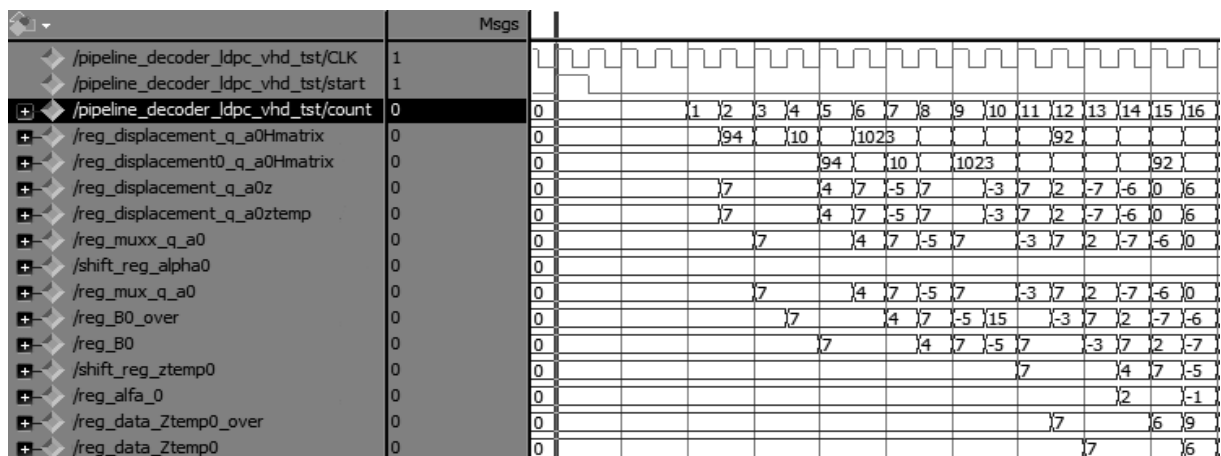


Рис. 5. Симуляція процесу декодування

З рисунка видно початок роботи конвеєра декодера та поступове підключення до роботи всіх складових конвеєра. Конвеєрна організація обчислень дозволила підвищити пропускну здатність декодера порівняно з неконвеєризованою версією в 40 разів.

Проведено процедуру розміщення розробленого декодера для ПЛІС Altera Stratix IV, яка показала, що декодер здатен працювати на тактовій частоті 240 МГц, а розрахункова пропускну здатність при 10 ітераціях становить понад 20 Мбіт/с.

За допомогою програмного забезпечення, написаного мовою програмування Java, проведено тестування роботи алгоритму мінімальної суми та його модифікованої версії. Типовий приклад виграшу в кількості ітерацій декодування для модифікованого алгоритму зображено на рис. 6.

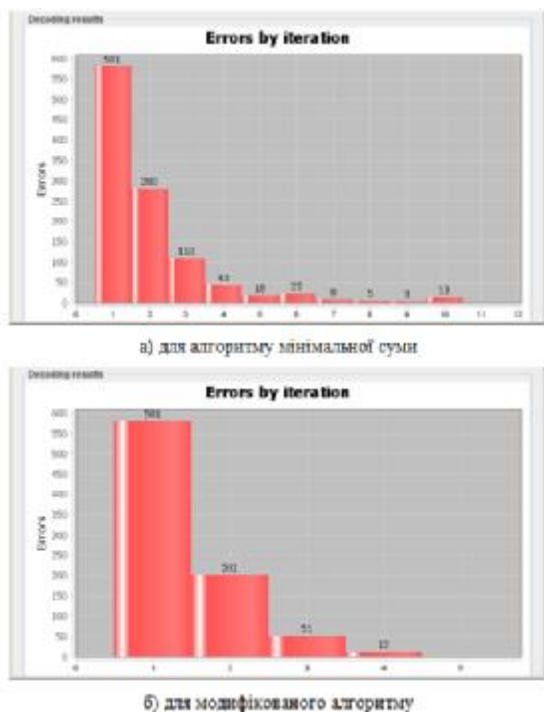


Рис. 6. Результати декодування за кількістю помилок за ітераціями

Для порівняння результатів використовувався однаковий пакет вхідних даних. При тестуванні прийому 100 пакетів виграш у кількості ітерацій становив 40 %. Кількість помилок перевірки парності у вхідному повідомленні, при якій модифікований алгоритм показував кращі результати декодування, становила для заданої матриці 30-40 % від загальної кількості перевірок парності. Це значення відповідає співвідношенню сигнал/шум приблизно від 4.5 дБ.

Висновки. У статті представлено метод побудови конвеєрної архітектури для нерегулярних LDPC-кодів з довільним розташуванням значущих елементів. Забезпечується можливість застосування конвеєра для таких МПП, побудова такого конвеєра з урахуванням особливостей алгоритму декодування мінімальної суми. Також представлено модифікацію алгоритму декодування, що дозволяє зменшити кількість ітерацій для декодування за умови наявності відносно невеликої кількості помилок у вхідних даних. Проведено реалізацію декодера для мікросхеми ПЛІС, яка підтвердила коректність запропонованих положень.

Список літератури

1. Gallager R. G. Low-density parity check codes / R. G. Gallager // IRE Transaction Info. Theory, IT-8. – 1962. – 21–28 January.
2. Prasad R. New horizons in mobile and wireless communications / R. Prasad, A. Mihovska. – 2009. – Vol. I. – P. 105.
3. Shirani-Mehr H. A reduced routing network architecture for partial parallel LDPC decoders / H. Shirani-Mehr, T. Mohsenin, B. Baas // 2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR). – 2011. – P. 2192-2196.
4. Karkooti M. Semi-parallel architectures for real-time LDPC coding: a thesis submitted in partial fulfillment of the requirements for the degree master of science / Marjan Karkooti. – Rice University, Houston, Texas, 2004. – 87 p.
5. Guilloud F. Generic architecture for LDPC codes decoding: PhD thesis / Frederic Guilloud. – ENST Paris. – 2004, July. – 200 p.
6. Musiyenko M. Reconfigurable partially parallel LDPC-decoder based on FPGA using quad-port block memory / M. Musiyenko, Y. Krainyk., O. Denysov // Conference Proceedings of IEEE ELNANO-2015. – 2015. – P. 498–503.
7. Крайник Я. М. Підвищення ефективності використання пам'яті частково паралельного ldpc-декодера / Я. М. Крайник // Вісник Черкаського державного технологічного університету. – 2014. – С. 10–14. ISSN 2306-4455.
8. Fossorier M. P. C. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation /

- M.P.C. Fossorier, M. Mihaljević, H. Imai // IEEE Transactions on Communications. – 1999. – Vol. 47, Issue 5. – P. 673-680.
9. Mohsenin T. Algorithms and architectures for efficient low density parity check (LDPC) decoder hardware: dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Electrical and Computer Engineering / Tinoosh Mohsenin. – University of California, Davis. – 2010. – 122 p.
- References**
1. Gallager, R. G. (1962) Low-density parity check codes. *IRE Transaction Info. Theory, IT-8*, 21–28 January.
 2. Prasad, R. and Mihovska, A. (2009) New horizons in mobile and wireless communications, (I), p. 105.
 3. Shirani-Mehr, H., Mohsenin, T. and Baas, B. (2011) A reduced routing network architecture for partial parallel LDPC decoders. *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 2192-2196.
 4. Karkooti, Marjan (2004) Semi-parallel architectures for real-time LDPC coding: a thesis submitted in partial fulfillment of the requirements for the degree master of science. Rice University, Houston, Texas, 87 p.
 5. Guilloud, F. (2004) Generic architecture for LDPC codes decoding: PhD thesis, ENST Paris. July, 2004, 200 p.
 6. Musiyenko, M., Krainyk, Y. and Denysov, O (2015) Reconfigurable partially parallel LDPC-decoder based on FPGA using quad-port block memory. *Conference Proceedings of IEEE ELNANO-2015*, pp. 498-503.
 7. Krainyk, Y. M. (2014) Rising efficiency of memory usage for partial parallel ldpc decoder. *Visnyk Cherkaskogo derzhavnogo tehnologichnogo universytety*, pp. 10–14. ISSN 2306-4455 [in Ukrainian].
 8. Fossorier, M.P.C., Mihaljević, M. and Imai, H. (1999) Reduced complexity iterative decoding of low-density parity check codes based on belief propagation. *IEEE Transactions on Communications*, 47 (5), pp. 673-680.
 9. Mohsenin, T. (2010) Algorithms and architectures for efficient low density parity check (LDPC) decoder hardware / By Tinoosh Mohsenin // Dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Electrical and Computer Engineering, University of California, Davis. – 2010. – 122 p.

Ya. M. Krainyk, postgraduate student, codebreaker7@ukr.net

O. O. Denysov, applicant, maildenisov@gmail.com

Petro Mohyla Black Sea State University
68 Desantnykiv str., 10, Mykolaiv, Ukraine

PIPELINED ARCHITECTURE OF LDPC-DECODING FOR FPGA USING MODIFIED MINIMAL SUM ALGORITHM

The method for construction of pipelined architecture of partially parallel LDPC-decoder (Low Density Parity Check) based on FPGA (Field Programmable Gate Array) is presented in the article.

The article aims at increasing of operation speed of decoders of irregular LDPC-codes based on arbitrary parity check matrices due to the development of the method for construction of pipelined architecture of partially parallel decoder and modification of the algorithm of minimal sum decoding.

The construction of architecture provides preliminary processing of parity check matrix in order to avoid access collision during decoding. Proposed architecture implements modified algorithm for minimal sum decoding. This algorithm allows to decrease overall number of decoding iterations provided the availability of relatively small number of errors in input data and to increase the throughput. Developed statements provide significant increasing in decoder throughput.

Keywords: LDPC-decoder, parity check matrix, minimal sum algorithm.

Стаття надійшла до редакції 18.12.2015 р.

*Рецензенти: М. П. Мусієнко, д.т.н., професор,
В. М. Рудницький, д.т.н., професор*