

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Кваліфікаційна наукова
праця на правах рукопису

Цаволик Тарас Григорович

УДК 004.75

ДИСЕРТАЦІЯ

МЕТОДИ ТА ЗАСОБИ ПЕРЕДАЧІ ДАНИХ В БЕЗПРОВІДНИХ СЕНСОРНИХ
МЕРЕЖАХ НА ОСНОВІ МОДУЛЯРНИХ КОРИГУЮЧИХ КОДІВ

05.13.05 – комп'ютерні системи та компоненти

05 «Технічні науки»

Подається на здобуття наукового ступеня кандидата технічних наук

Дисертація містить результати власних досліджень. Використання ідей,
результатів і текетів інших авторів мають посилання на відповідне джерело


Цаволик Т. Г.

(підпис, ініціали та прізвище здобувача)

Науковий керівник:
Яцків Василь Васильович,
доктор технічних наук, доцент

Ідентичність всіх примірників дисертації
ЗАСВІДЧУЮ:
Т.в.о.вченого секретаря спеціалізованої
вченої ради




Мельник А. М.

Тернопіль – 2018

АНОТАЦІЯ

Цаволик Т. Г. Методи та засоби передачі даних в безпроводних сенсорних мережах на основі модулярних коригуючих кодів. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.05 – комп'ютерні системи та компоненти. – Тернопільський національний економічний університет, Міністерство освіти і науки України, Тернопіль, 2018.

Дисертація присвячена вирішенню науково-прикладної задачі підвищення надійності передачі даних у безпроводних сенсорних мережах шляхом розробки методів та засобів завадостійкого кодування даних на основі модулярної арифметики.

У першому розділі проведений аналіз впровадження безпроводних сенсорних мереж (БСМ). БСМ складається з великої кількості вузлів, які спілкуються між собою за допомогою безпроводного каналу зв'язку. Сенсорні вузли можуть розташовуватись на далеких відстанях від спостережного пункту.

Основними вимогами до вузлів БСМ є мінімальне енергоспоживання, що забезпечить довготривалий термін роботи вузла. У БСМ є різні типи сенсорних вузлів, які використовуються для вимірювання тиску, рівня шуму, температури, вологості та інші. Також сенсорні вузли можуть бути розташовані не тільки на поверхності, а й під землею та під водою.

Проаналізовано методи та алгоритми, які використовуються для підвищення надійності передачі даних у БСМ. Встановлено, що сучасні методи не забезпечують необхідну надійність передачі даних або мають високу надлишковість. У БСМ втрати пакетів відбуваються у результаті перешкод у каналі зв'язку, помилок передачі, черг пакетів, зіткнення пакетів, відмов вузлів (через виснаження енергії) або з інших непередбачуваних причин. Крім того, через малу відстань передачі (100 - 400 м) дані проходять велику кількість ретрансляцій, що в свою чергу, призводить до збільшення кількості можливих помилок, які можуть стати причиною втрати пакетів.

Другий розділ присвячено розробці методів підвищення надійності передачі даних у безпроводних сенсорних мережах на основі коригуючих кодів системи залишкових класів та модулярних коригуючих кодів.

Удосконалено коригуючі коди системи залишкових класів за рахунок вибору спеціальної системи модулів. У цьому методі запропоновано розширену систему модулів, яка зберігає переваги спеціальної системи модулів і при цьому забезпечує необхідну їх кількість. Методи виявлення та виправлення помилок з використанням коригуючих кодів системи залишкових класів (СЗК) можна поділити на дві групи: виправлення помилок на основі обчислення проекції числа та виправлення помилок на основі обчислення синдрому. Проведені дослідження цих двох методів виявлення та виправлення помилок на основі коригуючих кодів СЗК показали, що реалізація пристроїв для виявлення та виправлення помилок на програмованих логічних інтегральних схемах (ПЛІС) з використанням запропонованої розширеної спеціальної системи модулів зменшує апаратні затрати на 26% для методу обчислення проекцій і на 10% для методу обчислення синдрому. Також швидкодія декодера підвищується на 18% для методу обчислення проекцій і на 9% для методу обчислення синдрому.

Запропоновано метод формування перевірочних символів у коригуючих кодах системи залишкових класів. Суть даного методу полягає в тому, що послідовність бітів, яку потрібно передати поділяється на k частин по 4 або 8 біт. Оскільки значення тетрад або байтів у позиційній системі числення (ПСЧ) є меншими за відповідні модулі, то отримані частини будуть залишками. Оскільки дане повідомлення подане залишками, то його не потрібно переводити з ПСЧ в СЗК. Для обчислення перевірочних символів повідомлення перетворюється в ПСЧ і обчислюється залишок по перевірочному модулю. У результаті отримуємо кодове слово, яке складається з інформаційних та перевірочних символів. Проведено експериментальні дослідження апаратних затрат та часу обчислення перевірочних символів при різній розрядності вхідних даних у коригуючих кодах СЗК та в запропонованому методі

формування перевірочних символів. Запропонований метод забезпечує зменшення апаратних затрат в середньому на 20% і залежить від розрядності вхідних повідомлень та підвищення швидкодії за рахунок того, що дане повідомлення не потрібно переводити з ПСЧ в СЗК і виконувати зворотне перетворення.

Розроблено метод виправлення помилок у двох інформаційних символах на основі модулярних корегуючих кодів з одним перевірочним символом. Виправлення помилок базується на послідовному розв'язку діофантових рівнянь з двома невідомими. Даний метод виявлення та виправлення помилок дозволяє збільшити швидкість коду приблизно на 20% та зменшити надлишковість модулярного коригуючого коду.

Отримав подальший розвиток двовимірний метод виправлення пакетів помилок на основі модулярного коригуючого коду. Процес кодування двовимірним кодом відбувається наступним чином: дані поступають у внутрішній кодер, де здійснюється обчислення перевірочних символів для кожного рядка, і записуються у буфер. З буфера кодові слова зчитуються і поступають у зовнішній кодер, в якому проводиться обчислення перевірочних символів по стовпцях. Виявлення помилки відбувається на основі аналізу синдрому, який обчислюється по рядках та по стовпцях. Якщо в таблиці синдром по рядках і стовбцях дорівнює нулю, то помилки немає, інакше помилка наявна в рядках і стовбцях. Якщо враховувати те, що нам відоме місце розташування помилок, то модулярний код може виправляти дві помилки при розміщенні спотворених символів у двох рядках і одному стовпці або в двох стовпцях і одному рядку. Але даний код не може виправити пакет помилок з шести символів, які розміщуються на перетині трьох рядків і стовпців.

У третьому розділі розроблено алгоритми виявлення та виправлення помилок на основі модулярних коригуючих кодів та структури кодерів/декодерів. Розроблено алгоритм для пошуку взаємно простих коефіцієнтів для обчислення перевірочних символів у модулярному коригуючому коді.

Розроблено алгоритм виправлення помилок у модулярних коригуючих кодах. Даний алгоритм дозволяє виправляти помилки в двох інформаційних символах з використанням одного перевірного символу.

Розроблено алгоритм виявлення та виправлення пакетів помилок на основі двовимірних модулярних коригуючих кодів, який забезпечує виправлення пакетів помилок.

У четвертому розділі розроблено структури та реалізовано на ПЛІС кодери/декодери завадостійкого кодування даних на основі коригуючих кодів системи залишкових класів та модулярних коригуючих кодів. Розроблено пристрій виправлення помилок на основі коригуючих кодів СЗК та проведено експериментальні дослідження при їх реалізації на ПЛІС у програмному середовищі Quartus. Реалізований на ПЛІС кодер формування коригуючих кодів СЗК та проведено експериментальні дослідження порівняння відомого методу із запропонованим методом формування коригуючих кодів СЗК.

Розроблений адаптивний протокол передачі даних на основі модулярних коригуючих кодів. Пакет даних відправляється з одним перевіряючим символом, який виявляє помилки в даних, і якщо він виявляє помилку то відправляється запит на передачу наступного перевіряючого символу, що дозволяє виправити помилку в повідомленні.

Ключові слова: безпроводні сенсорні мережі, кодування, декодування, модулярні коригуючі коди, система залишкових класів, коригуючі коди, спеціальна система модулів, модулярна арифметика, двовимірний код, пакет помилок.

ПЕРЕЛІК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковано основні наукові результати дисертації

1. Цавольк Т. Г., Яцкив В. В. Метод исправления ошибок на основе модулярных корректирующих кодов // Физика, математика, информатика. Вестник Брестского государственного технического университета. Брест, 2015. № 5 (850). С. 36-38.

2. Цаволик Т. Г., Яцків В. В. Метод формування корегувальних кодів у системі залишкових класів // Науковий вісник НЛТУ України. 2017. Вип. 27 (3). С. 191–194.

3. Яцків В. В., Цаволик Т. Г. Двовимірні коректуючі коди на основі модулярної арифметики // Вісник Хмельницького національного університету. Технічні науки. 2015. № 4 (227). С. 144 – 148.

4. Sieck, V. Yatskiv, A. Sachenko, T. Tsavolyk. Two-Dimensional Error Control Based on Modular Corrective Codes. International Journal of Computing, 2015. Vol. 14, Issue 4. P. 208-213.

5. Цаволик Т. Г. Корегуючі коди в системі залишкових класів зі спеціальними модулями // Вимірювальна та обчислювальна техніка в технологічних процесах. Технічні науки. 2016. № 3. С. 100 – 104.

6. Цаволик Т. Г. Яцків В. В. виправлення пакетів помилок на основі модулярного коригуючого коду // Науковий вісник НЛТУ України. 2018. Вип. 28(2). С. 155–158.

7. Патент України на корисну модель № 117493, МПК (2006) H04J 13/00 H04W84/18 (2009.01). Спосіб формування корегуючих кодів в системі залишкових класів / В. В. Яцків, Т. Г. Цаволик, Н. Г. Яцків. – № u201700839; заявл. 30.01.2017; опубл. 26.06.2017. – Бюл. № 12/2017.

Наукові праці, які засвідчують апробацію матеріалів дисертації

8. Yatskiv V. Tsavolyk T., Hu Zhengbing. Multiple Error Detection and Correction Based on Modular Arithmetic Correcting Codes. Proceedings of the 8-th 2015 IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS'2015, (Warsaw, Poland, September 24-26, 2015). Warsaw. 2015. Volume 2. P. 850-854.

9. Yatskiv V., Tsavolyk T. Two-Dimensional Corrective Codes Based on Modular Arithmetic. Proceedings of the XIIIth International Conference the Experience of Designing and Application of CAD System in Microelectronics

CADSM' 2015. (Lviv-Polyana, Ukraine, February 24-27, 2015). Lviv-Polyana. 2015. P. 291-294.

10. Yatskiv V., Tsavolyk T., Sachenko A. Error Correction Technique Based on Modular Correcting Codes. Conference Proceedings IEEE 36th International Conference on Electronics and Nanotechnology (ELNANO), (Kyiv, Ukraine, April 19-21, 2016). Kyiv. 2016. P.362-364.

11. Sachenko A., Yatskiv V., Tsavolyk T. Modeling the Wireless Sensor Networks Using the Error Control Scheme. Proceedings of the 3rd IDAACS Symposium Wireless Systems within the IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS-SWS (Offenburg, Germany, September 26–27, 2016), Offenburg. 2016. P.122-126.

12. Yatskiv V., Tsavolyk T., Yatskiv N. The Correcting Codes Formation Method Based on the Residue Number System. Conference Proceedings of 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-2017) (Polyana-Svalyava, Ukraine, February 21-25, 2017). Polyana-Svalyava. 2017. P. 237-240.

13. Yatskiv V., Tsavolyk T. Improvement of Data Transmission Reliability in Wireless Sensor Networks on the Basis of Residue Number System Correcting Codes Using the Special Module System. Conference Proceedings, IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON) Igor Sikorsky Kyiv Polytechnic Institute, (Kyiv, Ukraine, May 29 – June 2, 2017). Kyiv. 2017. P. 890 – 893.

14. Yatskiv V., Tsavolyk T., Yatskiv N. Burst error-correcting codes based on modular correcting codes. Conference Proceedings of 14th International Conference Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET-2018) (Lviv-Slavske, Ukraine February 20-24 2018). Lviv. 2018. P. 389-392.

15. Цаволик Т.Г. Коректуючі коди системи залишкових класів зі спеціальною системою модулів. Матеріали V Всеукраїнської школи-семінару

молодих вчених і студентів «Сучасні комп'ютерні інформаційні технології» (м. Тернопіль, 22-23 травня 2015 р.). Тернопіль: ТНЕУ. 2015. С. 61-62.

16. Яцків В. В., Цаволик Т. Г. Метод завадостійкого кодування даних на основі модулярних коректуючих кодів. Матеріали V Міжнародної науково-практичної конференції молодих вчених «Інфокомунікації – сучасність та майбутнє» (м. Одеса, 29-30 жовтня 2015 р.). Одеса, ОНАЗ. 2015. С. 111-114.

17. Яцків В. В., Цаволик Т. Г., Стасюк І. І. Адаптивний метод контролю помилок в безпроводних сенсорних мережах. Прикладні науково-технічні дослідження: матеріали Міжнародної науково-практичної конференції (м. Івано-Франківськ, 5-7 квітня 2017 р.) Івано-Франківськ: Симфонія форте. 2017. С. 30.

ABSTRACT

Tsavolyk T.G. Methods and Means of Data Transmission in Wireless Sensor Networks Based on Modular Corrective Codes. – Qualification scientific work on the rights of manuscript.

Thesis is for obtaining Candidate of Technical Sciences degree in specialty 05.13.05 – Computer System and Components. Ternopil National Economic University, Ministry of Education and Science of Ukraine, Ternopil, 2018.

The dissertation is dedicated to solving the actual scientific and technical problem of increasing the reliability of data transmission in wireless sensor networks, through the development of methods and means and noise immunity coding on the basis of modular arithmetic.

In the first section, the analysis of the implementation of wireless sensor networks is accomplished (WSN). WSN consists of a large number of nodes that communicate with each other using a wireless communication channel. Sensor nodes can be located at long distances from the observation point.

The main requirements for the WSN nodes are the minimum power consumption, which will provide a long-term operating time of the nodes. There are

different types of sensor nodes in WSN that are used to measure pressure, noise, temperature, humidity, and others. Also, sensor nodes can be located not only on the surface but also underground and underwater.

The methods and algorithms that are used to improve the reliability of data transfer in the WSN are analyzed. It is established that modern methods do not provide the necessary reliability of data transmission or have a high redundancy. In WSN, packet loss occurs due to interruptions in the communication channel, transmission errors, packet queues, packet collisions, node failure (due to energy depletion), or other unforeseen reasons. In addition, due to the small transmission distance (100-400 m), data pass a large number of retransmissions, which in turn lead to increasing in the number of possible errors that could cause the loss of packets.

The second section is dedicated to the development of methods for increasing the reliability of data transmission in wireless sensor networks based on the corrective codes of the system of residual classes and modular corrective codes.

Corrective codes of the system of residual classes have been improved due to the choice of a special system of modules. In this method, an extended system of modules is proposed which preserves the advantages of a special system of modules and at the same time provides the necessary number of modules. Methods for detection and error correction by using correction code of residual number system (RNS) can be divided into two groups: the correction of errors based on projections of calculating the number of errors and corrections based on the calculation syndrome. The investigations of these two methods of detecting and correcting errors based on correction codes RNS has shown that the implementation of devices to detect and correct errors on programmable logic integrated circuits (FPGAs) using the proposed extended special system modules reduces hardware costs by 26% for the method of calculating projections and 10% for the method of calculating the syndrome. Also, the decoder's performance is increased by 18% for the prediction calculation method and by 9% for the calculation of the syndrome.

The method of formation of verification symbols in the corrective codes of the system of residue classes is proposed. The essence of this method is that the sequence

of bits that subjects the transmission is divided into k parts by 4 or 8 bits. Since the value of tetrads or bytes in the positional number system (PNS) is smaller than the corresponding modules, the resulting parts will be the remnants. Since this message is presented with the remnants, it is not necessary to convert it from the PNS to the RNS. To calculate the verification characters the message is converted to PNS and the remainder of the verification module is calculated. As a result, we get a codeword consisting of informational and verification characters. The experimental research hardware cost and time computation checking symbols at different discharges of bits of incoming data into correcting codes RNS is conducted and the method of forming checking symbols is proposed. The proposed method would reduce hardware costs by an average of 20%, depending on the bit of incoming messages and improving performance due to the fact that the messages need not be converted from PNS to RNS and perform an inverse transformation.

The method of correction of errors in two informational symbols on the basis of modular correction codes with one checking symbol is elaborated. The error correction is based on the sequential solution of Diophantine equations with two unknowns. This method of detecting and correcting errors allows you to increase the speed of the code by about 20% and reduce the redundancy of the modular correction code.

The two-dimensional method of correcting error packages on the basis of modular corrective code has been further developed. The process of encoding of the two-dimensional code is as follows: data enters the internal encoder, where the verification characters are calculated for each line and written to the buffer. From the buffer, the codewords are read and entered into an external encoder, in which the calculation of verification characters in the columns is performed. The detection of an error occurs on the basis of syndrome analysis, which is calculated in rows and columns. If there is the syndrome in the table in rows and columns equals zero, there are no errors, otherwise, the error is present in rows and columns. If we take into consideration the fact that we know the location of errors, then the modular code can correct two errors when placing distorted characters in two lines and one column or

in two columns and one line. But this code can't fix a six-character error package that is placed at the intersection of three rows and columns.

In the third section algorithms for detecting and correcting errors based on modular correction codes and encoder/decoder structures are developed. An algorithm for finding mutually simple coefficients for calculating checking characters in a modular corrected code is developed.

The algorithm of correction of errors in modular corrective codes is developed. This algorithm allows you to correct errors in two informational symbols using a single verification character.

The algorithm of detection and correction of error packages on the basis of two-dimensional modular corrective codes is developed, which provides correction of error packages.

In the fourth section, the structures and FPGA codecs/decoders of noise-dampening data coding based on corrective codes of the system of residual classes and modular corrective codes are developed. The error correction device based on the correction codes of the RNS was developed and experimental research on their implementation on the FPGA in the Quartus software environment was carried out. The coding of the correction codes of the RNS was implemented on the FPGA and experimental studies comparisons of the known method with the proposed method for the formation of corrective RNS codes were performed.

An adaptive data transfer protocol based on modular corrective codes is developed. A data package with one verification symbol that detects errors in the data is sent, and if it finds an error, then a request for the transmission of the next verification character is sent, which allows you to correct the error in the message.

Keywords: wireless sensor networks, coding, decoding, modular correction codes, residual number system, corrective codes, a special modular system, modular arithmetic, a two-dimensional code, error package.

LIST OF PUBLICATIONS BY THE SUBJECT OF DISSERTATION

Publications, in which the main scientific results of the dissertation are published

1. Tsavolyk T. G, Yatskiv V. V. Method of error correction on the basis of modular correction codes // Physics, mathematics, computer science. Bulletin of the Brest State Technical University. Brest, 2015. № 5 (850). P. 36-38.

2. Tsavolyk T. G., Yatskiv V. V. The method of formation of correction codes in the system of residual classes // Scientific herald of NLTU of Ukraine. 2017. Issue 27 (3). P. 191-194.

3. Yatskiv V. V, Tsavolyk T. G. Corrective Two-dimensional codes based on modular arithmetic // Herald of Khmelnytsky National University. Technical sciences. 2015. No. 4 (227). P. 144 - 148.

4. Sieck, V. Yatskiv, A. Sachenko, T. Tsavolyk. Two-Dimensional Error Control Based on Modular Corrective Codes. International Journal of Computing, 2015. Vol. 14, Issue 4. P. 208-213.

5. Tsavolyk T.G. Corrective codes in residual number system with special modules // Measuring and computing engineering in technological processes. Technical sciences. 2016. No. 3. P. 100-104.

6. Tsavolyk T.G., Yatskiv V.V. Error correction packet based modular correcting code // Scientific Herald NLTU Ukraine. 2018. Vip. 28 (2). Pp. 155-158.

7. Patent of Ukraine for Utility Model No. 117493, IPC (2006) H04J 13/00 H04W 84/18 (2009.01). Method of forming corrective codes in residue number system / V. V. Yatskiv, T. G. Tsavolik, N. G. Yatskiv. – № u201700839; stated. 01.30.2017; has published June 26, 2017 – Bull No. 12/2017

Publications certifying the approbation of the materials of the dissertation

8. Yatskiv V. Tsavolyk T., Hu Zhengbing. Multiple Error Detection and Correction Based on Modular Arithmetic Correcting Codes. Proceedings of the 8-th 2015 IEEE International Conference on Intelligent Data Acquisition and Advanced

Computing Systems, IDAACS'2015, (Warsaw, Poland, September 24-26, 2015). Warsaw. 2015. Volume 2. P. 850-854.

9. Yatskiv V., Tsavolyk T. Two-Dimensional Corrective Codes Based on Modular Arithmetic. Proceedings of the XIIIth International Conference the Experience of Designing and Application of CAD System in Microelectronics CADSM' 2015. (Lviv-Polyana, Ukraine, February 24-27, 2015). Lviv-Polyana. 2015. P. 291-294.

10. Yatskiv V. Tsavolyk T., Sachenko A. Error Correction Technique Based on Modular Correcting Codes. Conference Proceedings IEEE 36th International Conference on Electronics and Nanotechnology (ELNANO), (Kyiv, Ukraine, April 19-21, 2016). Kyiv. 2016. P.362-364.

11. Sachenko A., Yatskiv V., Tsavolyk T. Modeling the Wireless Sensor Networks Using the Error Control Scheme. Proceedings of the 3rd IDAACS Symposium Wireless Systems within the IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS-SWS (Offenburg, Germany, September 26–27, 2016), Offenburg. 2016. P.122-126.

12. Yatskiv V., Tsavolyk T., Yatskiv N. The Correcting Codes Formation Method Based on the Residue Number System. Conference Proceedings of 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-2017) (Polyana-Svalyava, Ukraine, February 21-25, 2017). Polyana-Svalyava. 2017. P. 237-240.

13. Yatskiv V., Tsavolyk T. Improvement of Data Transmission Reliability in Wireless Sensor Networks on the Basis of Residue Number System Correcting Codes Using the Special Module System. Conference Proceedings, IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON) Igor Sikorsky Kyiv Polytechnic Institute, (Kyiv, Ukraine, May 29 – June 2, 2017). Kyiv. 2017. P. 890 – 893.

14. Yatskiv V., Tsavolyk T., Yatskiv N. Burst error-correcting codes based on modular correcting codes. Conference Proceedings of 14th International

Conference Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET-2018) (Lviv-Slavske, Ukraine February 20-24 2018). Lviv. 2018. P. 389-392.

15. Tsavolyk T. G. Correcting code of residual number system with special system modules. Materials of V All-Ukrainian School-Workshop for Young Scientists and Students "Modern Computer Information Technologies" (Ternopil, May 22-23, 2015). Ternopil: TNEU. 2015. P. 61-62.

16. Yatskiv V. V., Tsavolyk T. G. The method of noise-proof data coding on the basis of modular correction codes. Materials of the V International Scientific and Practical Conference of Young Scientists "Infocommunications - Present and Future" (Odessa, October 29-30, 2015). Odessa, ONAT. 2015. P. 111-114.

17. Yatskiv V. V., Tsavolyk T. G., Stasiuk I. I. Adaptive method of error control in wireless sensor networks. Applied scientific and technical research: materials of the International scientific and practical conference (Ivano-Frankivsk, April 5-7, 2017) Ivano-Frankivsk: Symphony of forte. 2017. P. 30.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	17
ВСТУП	18
РОЗДІЛ 1. СУЧАСНИЙ СТАН ТА ПІДХОДИ ДО ПІДВИЩЕННЯ НАДІЙНОСТІ ПЕРЕДАЧІ ДАНИХ В БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖАХ	24
1.1 Аналіз стану розробки і впровадження безпроводних сенсорних мереж	24
1.2 Підходи до підвищення надійності передачі даних в безпроводних сенсорних мережах	34
1.3 Аналіз коригуючих кодів в системі залишкових класів	41
Висновки до першого розділу	51
РОЗДІЛ 2. МЕТОДИ ЗАВАДОСТІЙКОГО КОДУВАННЯ ДАНИХ В БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖАХ НА ОСНОВІ МОДУЛЯРНОЇ АРИФМЕТИКИ	53
2.1 Коригуючі коди системи залишкових класів зі спеціальною системою модулів	53
2.2 Метод формування перевірочних символів в коригуючих кодах системи залишкових класів	58
2.3 Метод виправлення помилок в двох інформаційних символах на основі модулярних коригуючих кодів з одним перевірочним символом	65
2.4 Метод виправлення пакетів помилок на основі модулярного коригуючого коду	71
Висновки до другого розділу	77
РОЗДІЛ 3. АЛГОРИТМИ ЗАВАДОСТІЙКОГО КОДУВАННЯ ДАНИХ НА ОСНОВІ МОДУЛЯРНИХ КОРИГУЮЧИХ КОДІВ	78
3.1 Алгоритм та програмна реалізація пошуку коефіцієнтів для побудови модулярних коригуючих кодів	78

	16
3.2 Алгоритм виявлення та виправлення помилок на основі модулярних коригуючих кодів	81
3.3 Алгоритм виявлення та виправлення пакетів помилок на основі двовимірних модулярних коригуючих кодів	85
3.4 Алгоритми виправлення помилок на основі коригуючих кодів системи залишкових класів	92
Висновки до третього розділу	96
РОЗДІЛ 4. СИНТЕЗ ПРИСТРОЇВ ЗАВАДОСТІЙКОГО КОДУВАННЯ ТА ПЕРЕДАВАННЯ ДАНИХ В БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖАХ НА ПРОГРАМОВАНИХ ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМАХ	98
4.1 Пристрій виправлення помилок на основі коригуючих кодів системи залишкових класів зі спеціальною системою модулів	98
4.2 Пристрій обчислення перевірочних символів в коригуючих кодах системи залишкових класів	107
4.3 Пристрій завадостійкого кодування/декодування даних на основі двовимірних модулярних коригуючих кодів	113
4.4 Адаптивний протокол передачі даних на основі модулярних коригуючих кодів	118
Висновки до четвертого розділу	128
ВИСНОВКИ	130
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	132
Додаток А Лістинг програми для знаходження взаємно простих коефіцієнтів	145
Додаток Б Декодер для виявлення та виправлення багатократних помилок	148
Додаток В Акти про використання результатів дисертаційного дослідження	150
Додаток Г Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації	154

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- БСМ – Безпроводна сенсорна мережа.
- СЗК – Система залишкових класів.
- ПСЧ – Позиційна система числення.
- CRT – Chinese remainder theorem (китайська теорема про залишки)
- FCS – Frame Check Sequence (контроль послідовного кадру).
- QoS – Quality of Services (якість обслуговування).
- MOS – MANTIS operation system (операційна система MANTIS).
- ARQ – Automatic repeat request (автоматична повторна передача).
- FEC – Forward Error Correction (пряме виправлення помилок).
- CRC – Cyclic redundancy check (циклічний надлишковий код).
- ПЛІС – Програмована логічна інтегральна схема.

ВСТУП

Актуальність роботи. Безпроводні сенсорні мережі (БСМ) є одним з найбільш перспективних напрямків розвитку розподілених обчислювальних систем для моніторингу та управління ресурсами і процесами. Безпроводні сенсорні мережі широко використовуються в системах екологічного, технічного та медичного моніторингу.

Для забезпечення необхідного рівня надійності передачі даних у БСМ, на відміну від безпроводних комп'ютерних мереж, необхідно враховувати обмеження, які накладаються низькими обчислювальними ресурсами вузлів та використанням автономного живлення. Крім того, додаткові ускладнення викликає режим ретрансляції пакетів, який використовується у більшості застосувань БСМ. Оскільки в БСМ часто використовують методи зменшення надлишковості трафіку шляхом стиснення даних, фільтрації корельованих даних, виключення надлишкових даних сенсорів, відповідно зростають вимоги до надійності передачі кожного пакету. Отже, для ефективної роботи БСМ необхідно забезпечити високу надійність і енергоефективність передачі даних при різних режимах роботи мережі.

Основним механізмом підвищення надійності передачі даних у БСМ є використання схем контролю помилок. Їх задачею є забезпечення надійного зв'язку в безпроводному каналі, в якому помилки виникають через перешкоди, завмирання та втрати бітової синхронізації. Це призводить до каналних помилок, які впливають на цілісність пакетів, що формують сенсорні вузли. До збільшення помилок також призводить низька потужність передавачів БСМ, що пов'язано з використанням автономного живлення.

Таким чином, крім методів підвищення надійності на фізичному рівні, які гарантують надійність передачі на рівні бітів, у БСМ також виникає необхідність застосування схем контролю помилок на каналному рівні, щоб забезпечити надійність на рівні передачі пакетів.

Використання в БСМ відомих завадостійких кодів (Ріда – Соломона, Боуза – Чоудхурі – Хоквінгема (БЧХ), згорткові коди та ін.), на відміну від модулярних коригуючих кодів, не дозволяє реалізувати адаптивні схеми виявлення помилок і тим самим підвищити надійність та зменшити надлишовість від застосування коригуючих кодів.

Фахівцями, які зробили значний теоретичний внесок у розвиток системи залишкових класів (СЗК), є: І. Я. Акушський, В. А. Торгашев, В. М. Амербаєв, Я. М. Николайчук, М. І. Червяков, В. А. Краснобаєв, В. В. Яцків, А. Омонді (A. Omondi), А. Мохан, (A. Mohan).

Отже, відомі методи завадостійкого кодування та передачі даних у БСМ не забезпечують належного ефективного вирішення зазначеної задачі. Тому розробка і дослідження методів завадостійкого кодування даних, що забезпечують підвищення надійності передачі даних у БСМ, а також створення спеціалізованих пристроїв обробки даних, є актуальною науково-прикладною задачею.

Зв'язок роботи з науковими програмами, планами, темами. Розробка основних положень дисертаційного дослідження здійснювалась відповідно до планів НДР, програм і договорів, що виконувалися у Тернопільському національному економічному університеті:

– НДР на тему «Методи та засоби побудови безпроводних мультимедійних сенсорних мереж на основі модулярної арифметики» (номер державної реєстрації 0112U007886);

– НДР на тему «Теоретичні основи та апаратні засоби підвищення продуктивності роботи безпроводних сенсорних мереж» (номер державної реєстрації 0117U000414).

Мета і задачі дослідження. Мета дисертаційної роботи – підвищення надійності передачі даних у безпроводних сенсорних мережах шляхом розробки та дослідження методів і засобів завадостійкого кодування даних на основі модулярної арифметики.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- 1) проаналізувати сучасний стан та шляхи удосконалення методів підвищення надійності передачі даних у безпроводних сенсорних мережах;
- 2) розробити метод виправлення помилок на основі модулярних коригуючих кодів;
- 3) розробити метод формування перевірочних символів у коригуючих кодах системи залишкових класів;
- 4) удосконалити коригуючі коди системи залишкових класів при їх реалізації на програмованих логічних інтегральних схемах;
- 5) розробити адаптивний протокол передачі даних на основі модулярних коригуючих кодів;
- 6) побудувати моделі та реалізувати на програмованих логічних інтегральних схемах пристрої завадостійкого кодування/декодування даних на основі розроблених методів;
- 7) провести дослідження ефективності запропонованих рішень у безпроводних сенсорних мережах.

Об'єкт дослідження – процеси обробки і передачі даних у безпроводних сенсорних мережах.

Предмет дослідження – методи та засоби завадостійкого кодування та передачі даних у безпроводних сенсорних мережах на основі модулярної арифметики.

Методи дослідження. При розв'язку задач у дисертаційній роботі використовувались: теорія інформації та кодування, методи теорії чисел, методи синтезу і аналізу цифрових схем при розробці пристроїв у системі залишкових класів, методи завадостійкого кодування при розробці модулярних коригуючих кодів.

Наукова новизна одержаних результатів.

1) Вперше розроблено метод виправлення помилок на основі модулярних коригуючих кодів з одним перевірочним символом, який, на відміну від відомих, базується на послідовному розв'язку діофантових рівнянь з двома

невідомими, що дозволило збільшити кодову відстань модулярного коригуючого коду та підвищити швидкість коду.

2) Вперше розроблено метод формування перевірочних символів у коригуючих кодах системи залишкових класів, у якому, на відміну від відомих, перетворення повідомлення в систему залишкових класів відбувається за рахунок поділу повідомлення на частини та вибору необхідної величини модулів, що дозволило підвищити швидкодію кодування і зменшити апаратні затрати на реалізацію пристроїв кодування.

3) Удосконалено коригуючі коди системи залишкових класів шляхом вибору спеціальної системи модулів, яка забезпечує зменшення апаратної складності та підвищує швидкодію реалізації кодерів/декодерів на програмованих логічних інтегральних схемах.

4) Отримав подальший розвиток двовимірний метод виправлення помилок, який базується на новому методі виправлення помилок на основі модулярних коригуючих кодів, що дозволило реалізувати виправлення пакетів помилок більшої довжини.

Практичне значення одержаних результатів полягає в розробці:

1) кодера обчислення перевірочних символів на основі запропонованого методу формування перевірочних символів у коригуючих кодах системи залишкових класів;

2) декодера виправлення помилок у двох інформаційних символах на основі модулярних коригуючих кодів з одним перевірочним символом;

3) декодера виправлення помилок на основі коригуючих кодів системи залишкових класів зі спеціальною системою модулів;

4) адаптивного алгоритму та протоколу передачі даних у безпроводних сенсорних мережах з використанням коригуючих кодів системи залишкових класів.

Теоретичні та практичні результати дисертаційної роботи використано і впроваджено у: ТОВ «Протекшин-Груп»; ТОВ «Інтелдім» для підвищення надійності передачі даних з безпроводних сенсорів; при виконанні науково-

дослідних робіт, які виконувались у Тернопільському національному економічному університеті; у навчальному процесі кафедр кібербезпеки та комп'ютерної інженерії Тернопільського національного економічного університету при проведенні лекційних та лабораторних робіт з курсів «Теорія інформації та кодування», «Проектування в середовищі Інтернет – речей», «Програмування комп'ютерних систем на програмованих логічних інтегральних схемах», а також при підготовці магістерських робіт зі спеціальностей «Комп'ютерна інженерія» та «Комп'ютерні науки».

Особистий внесок здобувача. Наукові положення, які містяться в дисертації, отримані здобувачем особисто. У роботах, опублікованих у співавторстві, здобувачу належать: [1, 10, 16] – запропоновано метод та розроблено алгоритм виправлення помилок на основі модулярних коригуючих кодів; [2, 7, 12] – метод формування коригуючих кодів у системі залишкових класів та реалізація кодера на ПЛІС; [13, 15] – коригуючі коди в системі залишкових класів зі спеціальною системою модулів; [3, 4, 9] – алгоритм та дослідження виправлення помилок у двовимірних коригуючих кодах; [6, 14] – метод виправлення пакетів помилок на основі модулярного коригуючого коду; [8] – структура декодера виправлення помилок на основі модулярних коригуючих кодів; [11] – моделювання схем контролю помилок з використанням модулярних коригуючих кодів; [17] – запропоновано адаптивний метод контролю помилок у безпроводних сенсорних мережах.

Апробація результатів дисертації. Результати роботи обговорювались і доповідались на міжнародних та всеукраїнських наукових конференціях, зокрема на: 8-th IEEE Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications: IDAACS'2015 (Warszawa, Poland, 2015); 3 rd IDAACS Symposium Wireless Systems within the IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS-SWS 2016 (Offenburg, Germany, 2016); 36th IEEE International Conference on Electronics and Nanotechnology: ELNANO (Kyiv, Ukraine, 2016); 14 th International Conference The Experience of

Designing and Application of CAD Systems in Microelectronics, CADSM – 2017 (Polyana-Svalyava (Zakarpattya), Ukraine 2017); 1 rd IEEE Ukraine Conference on Electrical and Computer Engineering, UKRCON – 2017 (Kyiv, Ukraine, 2017); 14 th International Conference Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET – 2018 (Lviv-Slavske (Zakarpattya), Ukraine, 2018); IV, V Всеукраїнській школі-семінарі молодих вчених і студентів «Сучасні комп'ютерні інформаційні технології» (ТНЕУ, м. Тернопіль, 2014, 2015); V Міжнародній науково-практичній конференції молодих вчених «Інфокомунікації – сучасність та майбутнє» (ОНАЗ, м. Одеса, 2015); V Міжнародній науково-технічній конференції молодих учених та студентів «Актуальні задачі сучасних технологій» (ТНТУ, Тернопіль, 2017); Міжнародній науково - практичній конференції «Прикладні науково-технічні дослідження» (Івано-Франківськ, 2017).

Публікації. За темою дисертації опубліковано 17 наукових праць, у тому числі: 7 статей у провідних фахових виданнях (одна з них – одноосібна), з них: 1 – у періодичному іноземному виданні, 5 – у фахових виданнях України, які внесені до міжнародних наукометричних баз, 1 патент України на корисну модель та 10 тез доповідей у збірниках матеріалів конференцій. 7 наукових праць індексовано в наукометричних базах *Web of Science* та *Scopus*.

Структура та обсяг дисертації. Дисертація складається зі вступу, чотирьох розділів, висновків, списку використаної літератури і додатків. Загальний обсяг роботи становить 156 сторінок, у тому числі: 131 сторінка основного тексту, 51 рисунок, 22 таблиці та 4 додатка. Список літератури нараховує 116 найменувань.

РОЗДІЛ 1

СУЧАСНИЙ СТАН ТА ПІДХОДИ ДО ПІДВИЩЕННЯ НАДІЙНОСТІ ПЕРЕДАЧІ ДАНИХ В БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖАХ

1.1 Аналіз стану розробки і впровадження безпроводних сенсорних мереж

Безпроводна сенсорна мережа (БСМ) – це самоконфігуруюча мережа з великою кількістю територіально розподілених датчиків для вимірювання інформації, обміну та спільної обробки [4, 5, 69, 92].

Безпроводний сенсорний вузол містить процесор низької потужності, оперативну та постійну пам'ять, радіомодуль, джерело живлення та сенсори.

Розвиток БСМ починався із військових застосувань, наприклад виявлення супротивника, спостереження за полем бою та інші [83]. На даний час БСМ широко використовуються в багатьох інших сферах, таких як сільське господарство, екологічний моніторинг, включаючи забруднення повітря та води, моніторинг стану здоров'я, структурний моніторинг та інше [51, 99]. БСМ також ефективно застосовуються в системах контролю й управління технологічними процесами та виробництвом.

Відмінною особливістю БСМ є те, що більшість вузлів не передають дані безпосередньо на базову станцію, а передають дані сусіднім вузлам. Окремий безпроводний вузол має дуже обмежені ресурси (обчислювальну потужність, пам'ять та зв'язок).

Проте застосування БСМ часто вимагає широкомасштабного розгортання, коли кількість даних, що вимірюються, обробляються, передаються в мережі та зберігаються, ростуть пропорційно розміру мережі. Тому важливою задачею при проектуванні та розгортанні БСМ є ефективне використання обмежених ресурсів вузлів.

Розробка сенсорних мереж залежить від багатьох чинників, які включають в себе [89]:

- відмовостійкість,
- масштабованість,
- вид програмного забезпечення,
- апаратні обмеження,
- споживання енергії.

Ці чинники розглядаються в роботах багатьох дослідників. Однак ні в одному з цих досліджень повністю не враховані всі чинники, які впливають на розробку сенсорних мереж. Вони важливі, оскільки служать в якості орієнтиру для розробки протоколу або алгоритмів роботи сенсорних мереж. Крім того, ці чинники можуть бути використані для порівняння різних моделей роботи БСМ.

1.1.1 Програмне забезпечення та вимоги до безпроводних сенсорних мереж

В якості програмного забезпечення безпроводні сенсорні вузли використовують: TinyOS, LiteOS, Contiki, MANTIS [41, 27, 43].

TinyOS – операційна система з відкритим вихідним кодом, яка призначена для безпроводних сенсорних мереж. У TinyOS є вбудована операційна система (ОС), яка написана на мові `nesC` і являє собою ряд взаємодіючих завдань і процесів. TinyOS розробляється консорціумом, який очолює Каліфорнійський університет в Берклі спільно з дослідницьким підрозділом компанії Intel [25, 41].

LiteOS – це відкрита, інтерактивна, UNIX-подібна операційна система, призначена для безпроводних сенсорних мереж. Завдяки засобам, що постачаються разом із LiteOS, ми можемо керувати однією або кількома безпроводними сенсорними мережами у стилі Unix, передаючи дані, встановлюючи програми, отримуючи результати або налаштовуючи сенсори. Також можна розробляти програми для вузлів і за допомогою безпроводного каналу поширювати їх на сенсорні вузли [17].

Contiki – компактна, багатозадачна операційна система для вбудованих систем, зокрема розробляється для використання в пристроях з обмеженими ресурсами (в основному пам'ять) та 8-бітові комп'ютери для вбудованих систем на мікроконтролерах [24].

MANTIS – це легка та енергоефективна операційна система. Вона має розмір 500 байт, що включає ядро, планувальник і мережевий стек. Ключова особливість операційної системи MANTIS (MOS) полягає в тому, що вона переноситься на декілька платформ, тобто ми можемо тестувати програми MOS на настільних комп'ютерах. Після цього додаток може бути перенесений на безпроводний вузол. MOS також підтримує віддалене управління сенсорними вузлами за допомогою динамічного програмування. MOS написано на мові програмування C [14, 15].

Основними вимогами до вузлів БСМ є мінімальне споживання енергії, що забезпечуватиме тривалий термін роботи вузла від автономного живлення, таких як батарейки чи акумулятори. У сенсорних вузлах радіотрансівер споживає приблизно 70% енергії, отже, основною задачею є зменшення енергозатрат на передачу даних [44].

Вимоги до безпроводних сенсорних вузлів [81]:

- мінімальні розміри самого вузла;
- мінімальні затрати енергії;
- робота з великою кількістю вузлів;
- автономність;
- адаптація до навколишнього середовища;
- низька вартість вузла.

Безпроводні сенсорні мережі, як правило, не тільки вимірюють інформацію про стан середовища, але і виконують первинну обробку даних та передають їх на сервер [4, 88, 92]. БСМ, які найчастіше використовуються в задачах моніторингу, використовують топологію кластерного дерева (рисунок 1.1) [5]. У цій топології сусідні вузли, які є на шляху до базової станції, стають джерелом інформації а також ретранслятором даних. Базова

станція приймає та передає дані по безпроводному каналу зв'язку (WI-FI, GSM, супутниковий канал) на віддалені сервера, що підключені до мережі Інтернет.

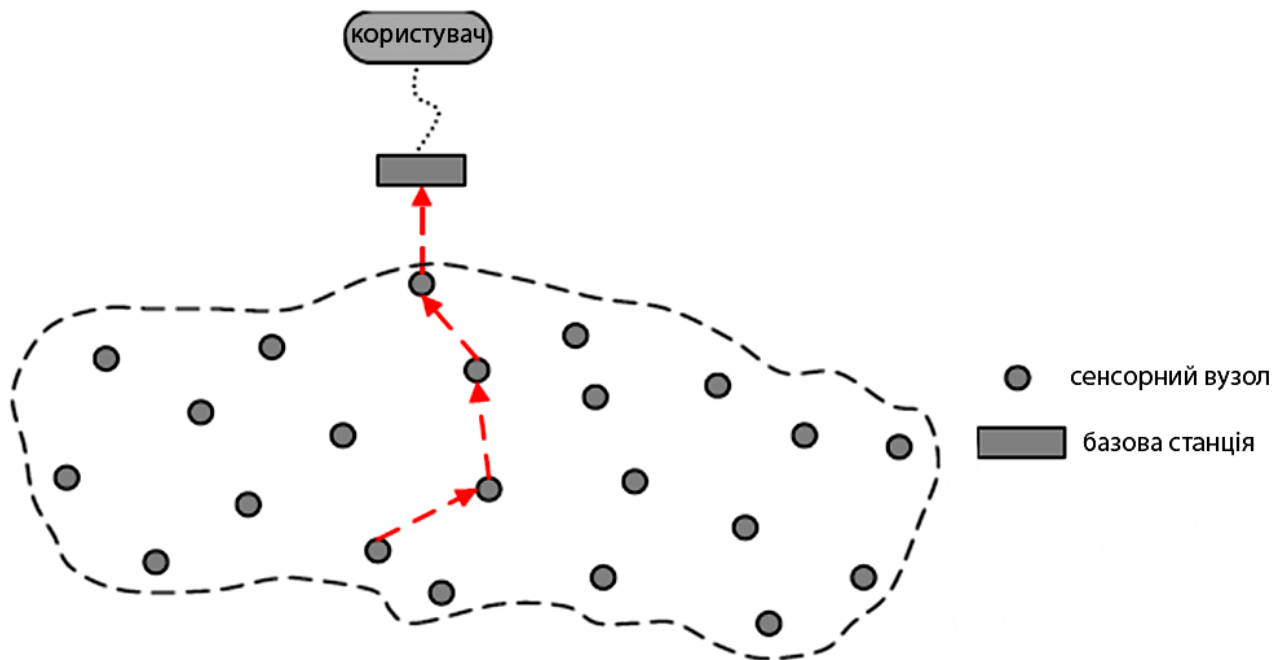


Рисунок 1.1 – Структура безпроводної сенсорної мережі

Більшість стандартів для безпроводної передачі даних, такі як: WI-FI, WiMAX, Bluetooth, HomeRF, ZigBee – використовують спільний частотний діапазон 2,4 ГГц, тим само створюючи один одному перешкоди.

На даний момент для розробки БСМ найбільш поширеним є стандарт IEEE 802.15.4 та стек протоколів ZigBee [35].

ZigBee – стек протоколів верхнього рівня, що використовує малопотужні радіомодулі, які базуються на стандарті IEEE 802.15.4. Даний стандарт ідеально підходить для БСМ, яким потрібно довгий період працювати від автономного джерела (батарейки, акумулятори) та висока безпека передачі даних при невеликій швидкості передачі даних.

Порівняльний аналіз характеристик стандартів для побудови безпроводних мереж приведений в таблиці 1.1.

Як видно з таблиці 1.1 найбільш оптимальним стандартом для побудови безпроводної сенсорної мережі є стек протоколів ZigBee, який базується на стандарті IEEE 802.15.4.

Таблиця 1.1 – Порівняльна характеристика стандартів безпроводних мереж

Стандарт	ZigBee (IEEE 802.15.4)	Wi-Fi (IEEE 802.11b)	Bluetooth (IEEE 802.15.1)
Частотний діапазон	2,4-2483 ГГц	2,4-2,483 ГГц	2,4-2,483 ГГц
Пропускна здатність, кбіт/с	250	11000	723,1
Розмір стеку протоколу, кбайт	32-64	більше 1000	більше 250
Час роботи від акумулятора, дні	100-1000	0,5-5	1-10
Максимальна кількість вузлів у мережі	65536	10	7
Діапазон дії, м	10-100	20-300	10-100
Область застосування	Віддалений моніторинг і управління	Передача мультимедійних даних (відео, картинки)	Заміщення дротової мережі

1.1.2 Класифікація та застосування безпроводних сенсорних мереж

Безпроводні сенсорні мережі складаються з різних типів сенсорів, наприклад, сейсмічних, сенсорів визначення магнітного поля, теплових, інфрачервоних, акустичних, які в змозі здійснювати найрізноманітніші вимірювання умов навколишнього середовища, зокрема, таких [20]:

- температура;
- вологість;
- автомобільний рух;
- тиск;
- склад ґрунту;
- рівень шуму;
- механічне навантаження;
- динамічні характеристики (швидкість, напрямок, розмір об'єкта).

Вузли БСМ використовуються для безперервного зондування, виявлення та ідентифікації подій. Серед найбільш поширених областей застосування БСМ необхідно виділити наступні:

- військове застосування [52, 55];
- моніторинг навколишнього середовища [1, 16, 18, 21, 30, 32, 36, 37, 39];
- охорона здоров'я [16, 39, 48, 56, 67];
- побудова інтелектуальних будинків [53];
- інші комерційні області.

Також можна розширити область застосування і додати ще такі категорії, як дослідження космічного простору та ліквідація наслідків стихійних лих і т. д.

Залежно від середовища передачі даних безпроводні сенсорні мережі поділяються на повітряні (наземні), підземні, підводні [82].

Наземні БСМ характеризуються великою кількістю вузлів, які можуть передавати дані на базову станцію. Часто сенсорні вузли можуть використовувати відновлювальні джерела живлення, наприклад, сонячні батареї та інші.

Підземні БСМ характеризуються набором вузлів сенсорів, які розміщені під землею або в закритому просторі, наприклад, печери, шахти та метро [2, 6]. Відмінність між підземними і наземними БСМ полягає у тому, що підземне середовище передачі характеризується значною втратою сигналу, і це потрібно враховувати при розробці протоколів передачі даних та розгортанні БСМ.

Вузли підводних БСМ розміщуються під водою і застосовуються для моніторингу підводної фауни, сейсмічного моніторингу, моніторингу обладнання та підводної робототехніки [3]. Підводні мережі використовують акустичні хвилі і мають низьку пропускну здатність, велику затримку сигналу і завмирання сигналу.

Мобільні БСМ. У мобільних БСМ вузли можуть переміщатися і взаємодіяти з фізичним середовищем. При розробці мобільних БСМ розробники стикаються з такими задачами, як навігація та управління рухомими вузлами, постійна підтримка зв'язності мережі, мінімізація споживання енергії при русі та іншими [22, 23, 84, 98, 104].

1.1.3 Структура безпроводної сенсорної мережі

Структура БСМ може будуватися на різних мережевих топологіях для мереж радіозв'язку, таких як:

- зіркоподібна мережа;
- сіткова мережа;
- гібридна або змішана мережа.

Зіркоподібна мережа [38]. Дана мережа може бути такою топологією зв'язку, що де б не існувала одна базова станція, вона буде відправляти або отримувати повідомлення різним віддаленим вузлам так, як показано на рисунку 1.2.

Віддаленим вузлам заборонено надсилати повідомлення одне до одного. Перевага цієї мережі для безпроводної сенсорної мережі полягає у простоті, здатності до збереження енергоспоживання віддаленого вузла до мінімуму. Недоліком мережі є те, що базова станція повинна бути серед радіопередачі.

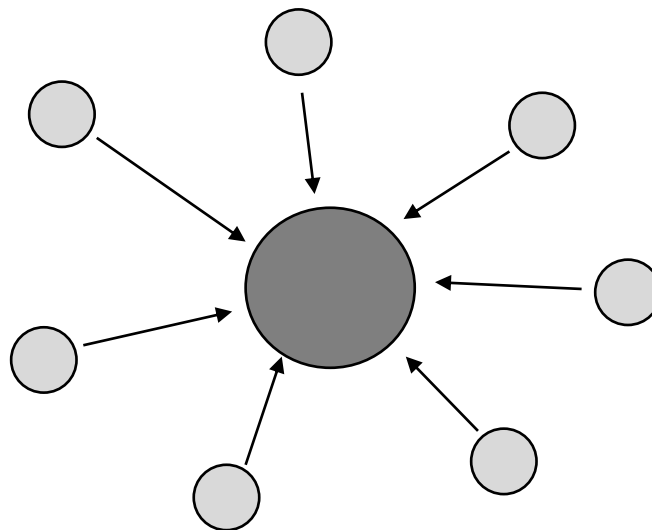

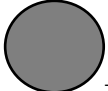


Рисунок 1.2 – Топологія зіркоподібної мережі

де:

-  – безпроводний сенсорний вузол;
-  – базова станція.

Сіткова мережа [38]. Дана мережа дозволяє передавати дані від одного вузла до іншого вузла в межах мережі, яка знаходиться в діапазоні радіопередачі, як показано на рисунку 1.3. Якщо вузол хоче відправити повідомлення іншому вузлу, який знаходиться поза діапазоном радіозв'язку, то для передачі буде використовуватися проміжний вузол. Перевага даної топології – низька надлишковість та висока масштабованість. Якщо окремому вузлові не вдається передати дані, то віддалений вузол зв'яжеться з іншим вузлом у своєму діапазоні і послідовно передасть повідомлення необхідному вузлу. Крім того, покриття мережі не обмежується діапазоном роботи окремих вузлів, мережа легко розширюється за рахунок додавання нових вузлів.

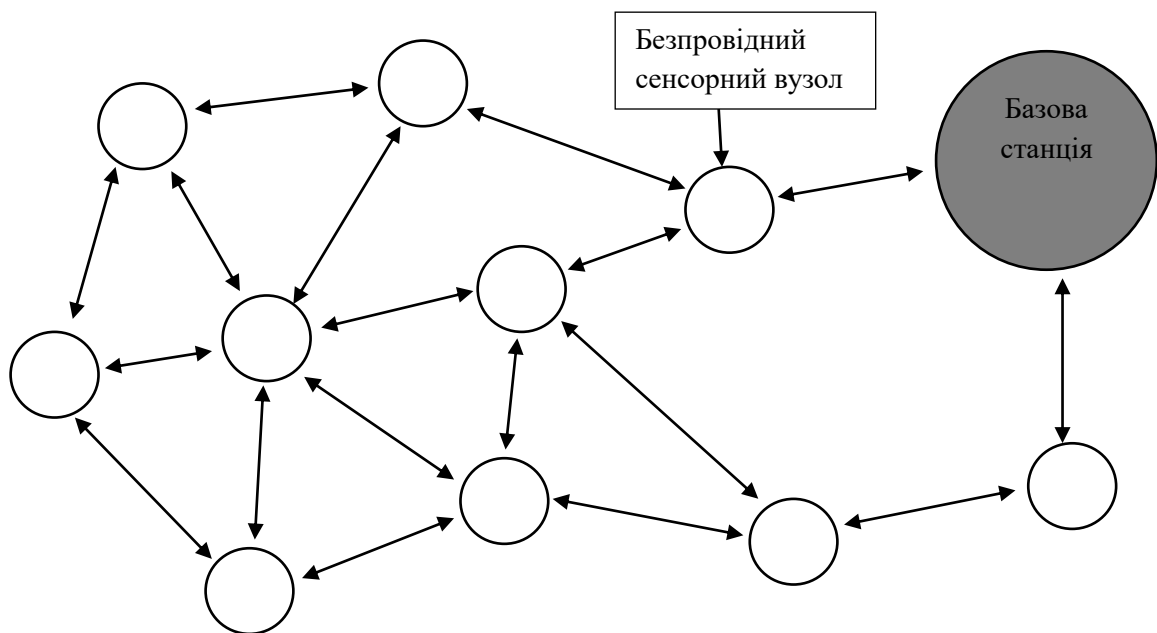


Рисунок 1.3 – Топологія сіткової мережі

Недоліком даної топології мережі є високі затрати енергії вузлами, які виконують обчислення та здійснюють ретрансляцію даних на наступні вузли,

Гібридна або змішана мережа (Star-Mesh) [38]. Гібрид між мережевою топологією зірки і сітки забезпечує надійність і універсальність мережевого

зв'язку, надаючи при цьому гнучкості у роботі та збереженні енергоспоживання вузлів до мінімуму, що приведено на рисунку 1.4.

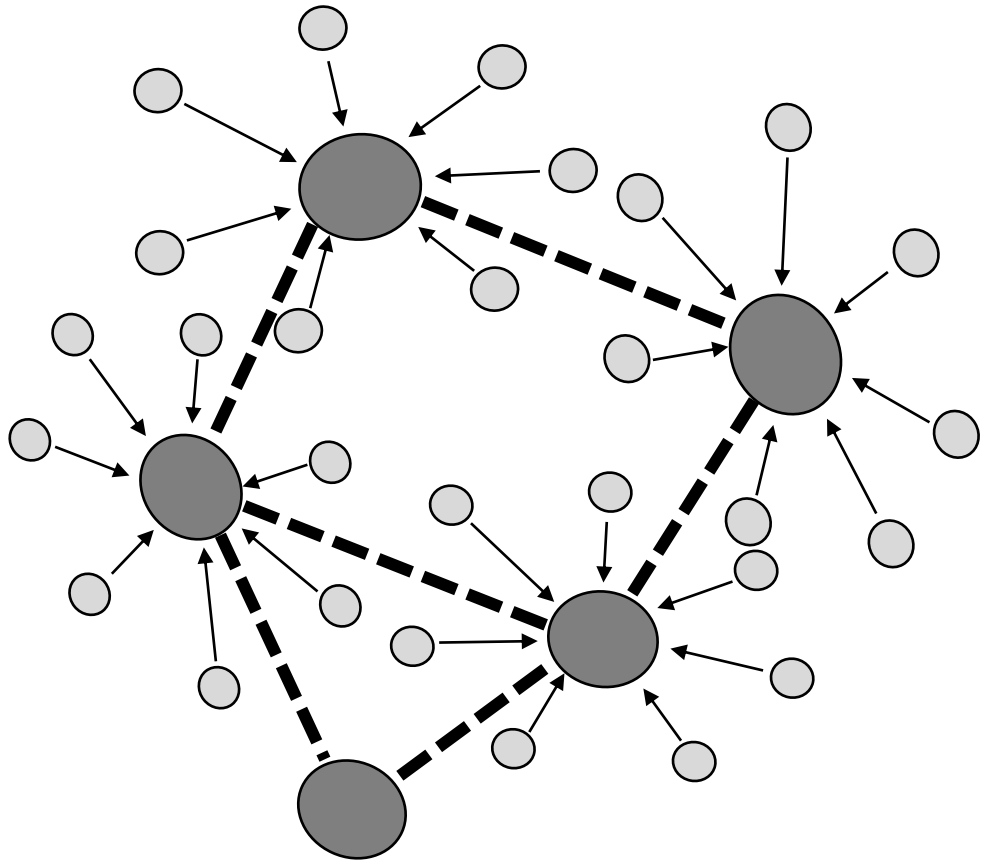



Рисунок 1.4 – Топологія гібридної мережі

де:

 - безпроводний сенсорний вузол;

 - базова станція.

У даній топології побудови мережі сенсорні вузли з малою потужністю не підтримують режим ретрансляції повідомлень. Проте вузли з розширеними можливостями, які як правило мають постійне живлення, можуть передавати повідомлення на інші вузли мережі.

Серед основних проблем, з якими стикаються дослідники при розробці БСМ, необхідно відзначити наступні [28, 33, 47]:

1. Обмежені ресурси. При реалізації та впровадженні БСМ необхідно враховувати обмежені ресурси:

- а) обмежена енергія;
- б) обмежена пам'ять;
- в) обмежена обчислювальна потужність;
- г) обмежена пропускна здатність.

2. Динамічна топологія та змінні умови навколишнього середовища. Залежно від застосування БСМ та середовища експлуатації топологія та характеристики мережі можуть змінюватися внаслідок збоїв зв'язку та відмови вузлів. Крім того, на вузли можуть додатково впливати високий рівень вологості, вібрації та інші умови, які можуть суттєво погіршити продуктивність мережі [28].

3. Різні вимоги до якості обслуговування. Велика різноманітність додатків БСМ може мати абсолютно різні вимоги та специфікації QoS. Якість обслуговування (QoS) БСМ – це точність між інформацією, що повідомляється вузлом, і реальною подією, яка виявляється сенсорами.

4. Збір даних. При високій щільності вузлів у топології мережі, дані окремих сенсорів можуть бути зайвими. Крім того, характер природних явищ є часовою кореляцією між кожним послідовним спостереженням вузла.

5. Надійність зв'язку. Порівнянно з провідними мережами можливості безпроводного зв'язку залежать від рівня перешкод та ймовірності бітових помилок [28].

6. Безпека. Безпека повинна бути обов'язковою особливістю в архітектурі БСМ, щоб зробити зв'язок безпечним від зовнішніх атак та вторгнення у відмову в обслуговуванні. БСМ мають особливі характеристики, які змінюють способи атак на даний тип мереж. Пасивні атаки здійснюються шляхом підслуховування передачі даних. Активні атаки передбачають захоплення вузлів або атаки маршрутизації.

7. Інтеграція з Інтернетом та альтернативними мережами. Для ефективного використання сенсорної мережі необхідно кілька сервісів, які у будь-який час надають доступ до будь-якої інформації, і таким чином можуть бути інтегровані з Інтернетом для отримання інформації кінцевими споживачами [10].

В роботі основна увага буде приділена підвищенню надійності передачі даних в БСМ шляхом розробки методів та засобів завадостійкого кодування даних на основі модулярних коригуючих кодів та надлишкової системи залишкових класів.

1.2 Підходи до підвищення надійності передачі даних в безпроводних сенсорних мережах

У БСМ втрати пакетів відбувається в результаті: перешкод у каналі зв'язку, помилок передачі, черг пакетів, зіткнення пакетів, відмов вузлів (через виснаження енергії) або з інших непередбачуваних причин. Крім того, через малу відстань (100 - 400 м) передачі, дані проходять велику кількість ретрансляцій, що, в свою чергу, призводить до збільшення кількості можливих помилок, які можуть стати причиною втрати пакетів. Таким чином, для успішного моніторингу середовища важливі дані, зібрані сенсорними вузлами, що повинні надійно бути доставлені в пункт обробки, забезпечуючи за необхідності відновлення втрачених даних.

Враховуючи імовірність та структуру помилок у безпроводних каналах, забезпечення надійної передачі даних з розподілених сенсорних вузлів у центральний вузол продовжує залишатися однією з основних проблем у сфері БСМ.

На даний час розроблена велика кількість протоколів для вирішення проблеми забезпечення надійності в бездротових сенсорних мережах.

У [45] для пошуку найкращої комбінації доступних варіантів забезпечення надійності в БСМ введена тривимірна (3D) еталона модель (рисунок 1.5), яка використовується для проведення аналізу та пошуку ефективної комбінації методів для вирішення проблеми досягнення надійності в БСМ.

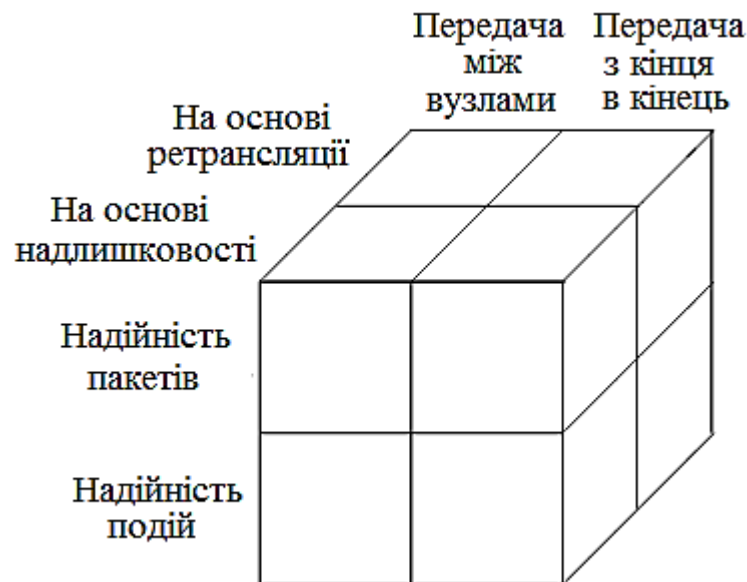


Рисунок 1.5 – Тривимірна еталонна модель для дослідження надійності в БСМ

Існуючі протоколи, які забезпечують надійну передачу, побудовані, як правило, з використанням одного з двох методів – повторної передачі (ARQ) або прямого виправлення помилок (FEC).

Ці методи забезпечують надійність, відновлюючи втрачені дані, використовуючи схему передачі «вузол – вузол» або «з кінця в кінець».

Повторна передача (ARQ) або пряме виправлення помилок (FEC). Повторна передача (ARQ) – це традиційний спосіб забезпечення надійності, де відправник після передачі пакету очікує на підтвердження відправленого пакета з наступного вузла на шляху до базової станції. Однак якщо відправник не отримує жодних підтверджень, він вважає, що відправлений пакет був втрачений. Таким чином, для забезпечення заданої надійності втрачений пакет

передається повторно. З іншого боку, замість повторної передачі всього пакета, підхід прямого виправлення помилок спрямований на виправлення лише втрачених або пошкоджених бітів усередині пакету. Відправник додає додаткову інформацію до пакету, яку приймач може використовувати для відновлення пакету, якщо деякі біти втрачені або пошкоджені. Ця додаткова інформація складається з надлишкового набору бітів, які кодуються за допомогою певного методу завадостійкого кодування.

Як повторна передача, так і пряме виправлення помилок можуть бути виконані за допомогою схеми «вузол – вузол» або «з кінця в кінець».

Схема «вузол – вузол» – лінійно-орієнтований механізм, тому що надійність повинна забезпечуватися на кожному окремому (між двома вузлами) каналі зв'язку. Інакше кажучи, в схемі «вузол – вузол» кожний інтервал від джерела до пункту призначення відповідає за надійне передавання даних. Аналогічним чином схему «з кінця в кінець» можна назвати механізмом, орієнтованим на зв'язок, де лише одне кінцеве значення (тобто, тільки джерело та пункт призначення) відповідає за забезпечення надійності, тоді як проміжні вузли просто передають пакети між джерелом і пунктом призначення.

1.2.1 Надійність на основі повторної передачі

У БСМ повторна передача є найбільш часто використовуваним механізмом для досягнення надійності передачі даних. Повторна передача також широко використовується в БСМ з обмеженими ресурсами для відновлення втрачених даних. Надійність на основі повторної передачі може бути забезпечена за допомогою схеми «з кінця в кінець» або «вузол – вузол». Механізм повторної передачі «з кінця в кінець» вимагає вихідного вузла, який генерує пакет для повторної передачі втраченого пакета, тоді як повторна передача «вузол – вузол» дозволяє проміжним вузлам здійснювати повторну передачу втрачених пакетів. У БСМ з багатоінтервальною маршрутизацією надійність на основі повторної передачі може бути досягнута через

підтвердження, коли відправник вимагає підтвердження своїх відправлених пакетів даних від наступного (сусіднього) одержувача на шляху до приймача.

Існує, як правило, два види механізмів підтвердження, а саме: явне підтвердження (eACK), яке передбачає також негативне підтвердження (NACK), та неявне підтвердження (iACK), як показано на рисунку 1.6. Механізм eACK спирається на спеціальне кероване повідомлення, яке вузол отримує після успішного отримання пакету, відправляє назад відправнику як квитанцію отриманого пакету, що показано на рисунку 1.6 (a) [45].

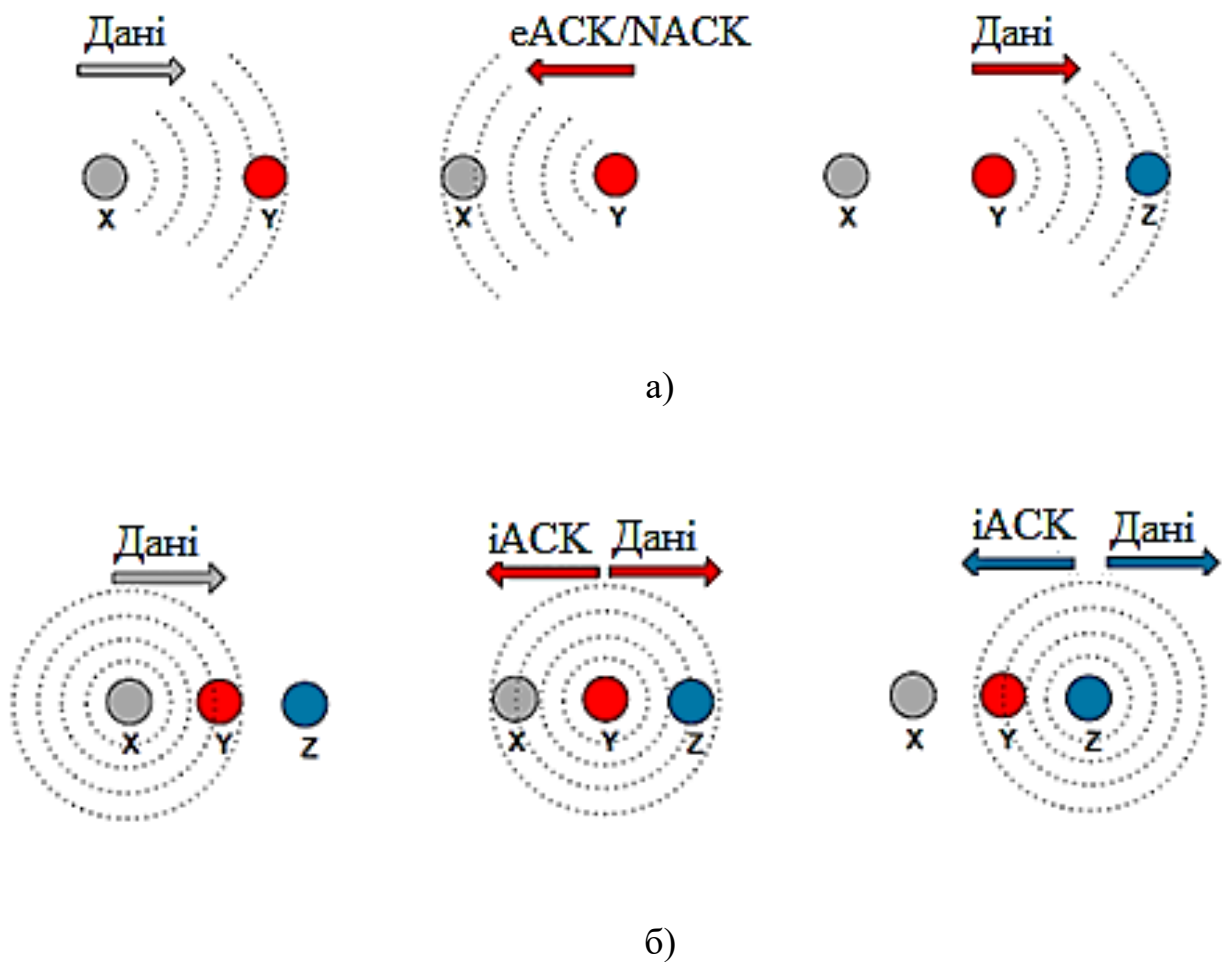


Рисунок 1.6 – Явний та неявний механізм підтвердження повторної передачі

Однак спеціальні керуючі повідомлення, передані в механізмах eACK та NACK, призводять до додаткових витрат енергії на передачу, що може бути

неможливим для більшості застосувань БСМ [45]. З іншого боку, в механізмі іАСК відправник після передачі пакета прослуховує канал й інтерпретує пересилання відправленого ним пакету наступним вузлом як квитанцію підтвердження (див. рисунок 1.6 б). Таким чином, іАСК використовує широкомовний характер безпроводного каналу, уникаючи додаткових витрат енергії на передачу та використання спеціальних контрольних повідомлень (eАСК / NACK).

1.2.2 Надійність на основі прямого виправлення помилок

У надійності на основі ретрансляції втрата бітів у пакеті трактується як втрата всього пакета і вимагає повторної передачі цього пакета для досягнення достовірної передачі даних. У концепції надійності на основі надлишковості, коли втрачені або пошкоджені біти всередині пакета можуть бути відновлені за допомогою методів заводостійкого кодування [96, 97]. Це значно зменшує накладні витрати на передачу, викликані повторною передачею всього пакета [98]. Враховуючи обмежені ресурси БСМ, використання методів заводостійкого кодування не тільки підвищить загальну надійність, але також збереже обмежені енергетичні ресурси безпроводних сенсорних вузлів.

Метод прямого виправлення помилок (FEC) використовується для виявлення та виправлення помилок на нижніх рівнях стеку протоколів у системах зв'язку та комп'ютерних мережах (наприклад, коди CRC у комп'ютерних мережах) [99]. Однак на верхніх рівнях (наприклад, транспортний рівень та рівень додатків) коригуючі коди потрібні для виправлення пакетів спотворених бітів. Підбираючи параметри оптимізації, заводостійке кодування також може бути реалізоване в обмеженому ресурсі БСМ.

Враховуючи обмежені функціональні характеристики безпроводних сенсорів, актуальною є проблема вибору коригуючих кодів для застосування в БСМ.

У [80] сформовані вимоги до коригуючих кодів в БСМ:

- низька складність алгоритмів кодування / декодування;
- низькі апаратні вимоги до реалізації алгоритмів (розрядність та тактова частота мікроконтролера, обсяг пам'яті);
- адаптивна зміна кількості перевірочних символів при зміні параметрів каналу.

З наведеного аналізу видно, що багато досліджень доводять переваги застосування в БСМ кодів Ріда – Соломона [34]. Однак використання кодів Ріда – Соломона унеможлиблює використання адаптивних схем контролю помилок. Оскільки при збільшенні / зменшенні кількості перевірочних символів необхідно перераховувати всі перевірочні символи [103]. Для вибору ефективної схеми контролю помилок та типу коригуючого коду потрібно також враховувати допустиму ймовірність помилки для заданої БСМ та характеристики каналу зв'язку.

1.2.3 Забезпечення якості обслуговування в безпроводних сенсорних мережах

Оскільки БСМ взаємодіють із середовищем вимірювання, їх характеристики відрізняються від інших традиційних мереж передачі даних. Відповідно, при розробці механізмів забезпечення якості обслуговування в БСМ необхідно враховувати [19]:

1) обмеження ресурсів, які включають енергію, пропускну здатність, пам'ять, розмір буферу, можливості обробки та обмежену потужність передавача. Серед них енергія – це основний параметр, оскільки вона серйозно обмежена в вузлах сенсорів, і часто неможливо замінити або заряджати акумулятор у сенсорних вузлах, які працюють у віддаленій або агресивній обстановці. Як результат, ці обмеження накладають суттєві вимоги на будь-які механізми підтримки QoS в БСМ, що унеможлиблює використання складних алгоритмів обчислень та протоколів сигналізації про стан мережі;

2) незбалансований трафік. У більшості додатків БСМ трафік формується від великої кількості вузлів сенсора до невеликої підмножини вузлів збору

даних. Отже, механізми QoS повинні бути розроблені для незбалансованого трафіку з обмеженим QoS;

3) надлишковість даних. БСМ характеризуються високою надлишковістю даних сенсорів. Однак, хоча надлишковість даних дає змогу зменшити вимоги до надійності даних та надійності передачі, вона вимагає багато енергії. Агрегація даних – це рішення для збереження надійності при зменшенні надлишковості даних, але цей механізм збільшує затримку й ускладнює QoS у БСМ;

4) динаміка мережі. Динаміка мережі може виникати внаслідок відмов вузлів, збоїв у безпроводному каналі, мобільності вузлів та змінах стану вузлів за рахунок використання схем управління енергоспоживанням. Високодинамічна мережа значно збільшує складність підтримки QoS;

5) енергетичний баланс. Для досягнення максимального терміну роботи мережі енергетичне навантаження повинно бути рівномірно розподілено між усіма вузлами, щоб енергія в одному безпроводному вузлі або невеликому наборі вузлів не вичерпувалася дуже швидко;

6) масштабованість. БСМ складається з сотень або тисяч вузлів щільно розподілених у місцевості. Отже, підтримка QoS, розроблена для БСМ, повинна бути спроможною масштабуватися до великої кількості вузлів. Підтримка QoS не повинна знижуватися при збільшенні кількості вузлів або їх щільності;

7) декілька вузлів збору (приймання) даних. Може існувати декілька вузлів збору даних, які встановлюють різні вимоги до мережі. Наприклад, один вузол може опитувати сенсорні вузли про температуру кожну хвилину, а інший вузол збору даних може цікавитись винятково високою температурною в іншій частині мережі. БСМ повинні мати можливість підтримувати різні рівні QoS, пов'язані з різними вузлами збору даних;

8) різний тип трафіку. Використання гетерогенних наборів датчиків створює проблеми для підтримки QoS. Наприклад, деякі додатки можуть використовувати різні типи сенсорів для контролю температури, тиску, вологості, звуку, бо таким чином використовуються різні режими зчитування з

цих сенсорів. Неоднорідне середовище робить підтримку QoS більш складною задачею.

9) критичність пакетів. Зміст даних відображає критичність реальних фізичних явищ і, отже, має різний пріоритет щодо якості додатків. Для диференціації важливості пакетів та створення пріоритетної структури необхідно передбачити механізми QoS.

1.3 Аналіз коригуючих кодів в системі залишкових класів

1.3.1 Система залишкових класів

Системи залишкових класів (СЗК) та надлишкова системи залишкових класів (НСЗК) привернули широку увагу дослідників при розробці високошвидкісних паралельних структур обробки сигналів [7, 54, 61, 63, 62, 70, 77, 85, 111]. НСЗК застосовується для виявлення та виправлення помилок в арифметичних операціях, цифрових процесорах, передачі даних [61].

Обчислювальна архітектура із самоперевіркою вперше була запропонована Уотсоном і Гастінгсом у [68]. Барсі та Маєстріні [13] розробили концепції дозволеного та забороненого діапазону, вивчаючи властивості корекції помилок в НСЗК. Дженкінс та інші успішно застосував алгоритми змішаного перетворення та базового розширення в НСЗК для реалізації цифрових фільтрів та контролю помилок у списках [63].

Теорія завадостійкого кодування, що використовує НСЗК, була запропонована Акушським і розвинута Кришною та ін. [42, 86]. У цій теорії кодування використовуються визначені умови, що обмежують коди НСЗК до лінійного простору [61]. Проте коди НСЗК відносяться до нелінійних кодів.

Важливою характеристикою коригуючого коду є ваговий розподіл, що дає уявлення про структуру коду. Оскільки код НСЗК (n, k) схожий на код Ріда – Соломона (РС), він також досягає мінімальної кодової відстані $d = n - k + 1$ за умови, що модулі коду НСЗК задовольняють певним умовам.

Також відомо, що розподіл вагових кодів НСЗК може бути апроксимований ваговим розподілом кодів RS, якщо всі модулі приймають значення, близькі до їх середнього значення [7].

На даний час виявлення та виправлення помилок при використанні кодів НСЗК залишається предметом досліджень багатьох вчених [70].

Концепція так званої «проекції» була введена Барсі та Маєстріні в [13], а також була досліджена теорія виправлення помилок на основі цих методів. Отже, кодування на основі НСЗК є перспективним методом виявлення та виправлення помилок у процесі передачі даних.

У роботах [49, 90, 91] представлено теоретичні основи розмежованої системи залишкових класів для порівняння та додавання чисел у системі залишкових класів і виконання високопродуктивних перетворень великорозрядних чисел з базису Радемахера у базис Крестенсона.

Використання розмежованої форми СЗК спрощує арифметичні операції у базисі Крестенсона та суттєво спрощує перехід з базису Радемахера в базис Крестенсона.

У роботах [93, 100] наведені аналітичні вирази для пошуку взаємно простих модулів, які утворюють досконалу форму СЗК. Використання даного методу зменшує обчислювальну складність при застосуванні китайської теореми про залишки для переведення чисел у позиційну систему числення за рахунок уникнення операції пошуку оберненого елемента за відповідним модулем.

У роботах [40, 95, 101] представлені теоретичні основи модифікованої досконалої форми системи залишкових класів та розроблений метод, який підбирає систему з трьох модулів, що утворюють модифіковану досконалу форму СЗК. Даний метод спрощує пошук системи модулів, які утворюють модифіковану досконалу форму СЗК при заданій кількості модулів. Модифікована досконала форма СЗК будується на основі розв'язку систем конгруенцій. У роботі [94] отримано формулу, яка визначає умову існування

відповідного набору модулів, та представлені аналітичні вирази для пошуку третього модуля і можливого діапазону обчислень.

Перевагою вказаних форм СЗК є те, що вони забезпечують зменшення обчислювальної складності при переведенні чисел із СЗК у позиційну систему числення. Проте дані форми СЗК не забезпечують необхідну кількість взаємно простих модулів для побудови коригуючих кодів.

Надлишкова СЗК широко досліджувалась тільки для проектування відмовостійких пристроїв обробки сигналів, а не для захисту від помилок у процесі передачі даних через канали зв'язку. Відповідно, теорія завадостійкого кодування з використанням НСЗК потребує додаткових досліджень.

Система залишкових чисел (СЗК) визначається вибором k цілих чисел p_i , ($i = 1, 2, \dots, k$), які називають модулями [86, 8]. Якщо всі модулі є попарно взаємно простими числами, то будь-яке ціле число X в діапазоні $[0, P_k)$, де

$P_k = \prod_{i=1}^k p_i$, може бути однозначно представлено послідовністю залишків b_i :

$$X \Leftrightarrow (b_1, b_2, \dots, b_k),$$

де $b_i = X \bmod p_i$, $i = 1, 2, \dots, k$,

b_i є залишками числа X від ділення на p_i для $i = 1, 2, \dots, k$. При цьому $[0, P_k)$ – інформаційний діапазон або робочий (дозволений) діапазон X .

Ціле число X у діапазоні $[0, P_k)$ може бути відновлено з k залишків (b_1, b_2, \dots, b_k) , використовуючи китайську теорему про залишки [86].

Зокрема, згідно з КТЗ для будь-якого заданого кортежу (b_1, b_2, \dots, b_k) , де $0 \leq b_i < p_i$, $i = 1, 2, \dots, k$, можна знайти значення X із залишків, використовуючи [86, 50]:

$$X = \left(\sum_{i=1}^k b_i \cdot M_i \cdot \delta_i \right) \bmod P_k,$$

де $M_i = \frac{P_k}{p_i}$, $\delta_i = M_i^{-1} \bmod p_i$.

Щоб отримати можливість виявлення та виправлення помилок у повідомленні, поданому в системі залишкових класів, необхідно ввести додаткові модулі.

Надлишкова СЗК отримується додаванням певного числа $r = (n - k)$ додаткових модулів $(p_{k+1}, p_{k+2}, \dots, p_{k+n})$ до раніше вибраної системи модулів. У результаті формується НСЗК – код з n додатних попарно взаємно простих модулів.

Тепер ціле число X в діапазоні $[0, P_k)$ виражається послідовністю n залишків по модулях (p_1, p_2, \dots, p_n) :

$$X \Leftrightarrow (b_1, b_2, \dots, b_n).$$

Отже, інтервал $[0, P_k)$ називається дозволеним, а інтервал $[P_k, P_n)$, сформований із додаткових модулів r , називається забороненим діапазоном (рисунок 1.7).



Рисунок 1.7 – Робочий та заборонений діапазони надлишкової СЗК

Робочий діапазон є обчислювальним діапазоном системи залишкових класів, тоді як заборонений діапазон використовується для виявлення та виправлення помилок [31, 9].

Нехай X ціле число, яке знаходиться в межах дозволеного діапазону, якщо залишки $(b_1, b_2, \dots, b_k, b_{k+1}, \dots, b_n)$ пошкоджені, то X буде перетворено в $X' = X + E$, де E – величина помилки. X' завжди буде знаходитись в забороненому діапазоні, якщо $p_{k+1}, p_{k+2}, \dots, p_{k+n} > p_i \quad \forall_i \leq k$.

Таким чином, перевіряючи значення X' , помилки в залишках можуть бути виявлені, а потім виправлені.

Коригуюча здатність надлишкової СЗК визначається кількістю введених додаткових модулів.

При додаванні r додаткових модулів надлишкова СЗК здатна виявляти r і виправляти помилки в $\lfloor r/2 \rfloor$ залишках [11].

Виявлення та виправлення помилок у залишках виконується, як правило, упродовж трьох послідовних етапів:

- 1) виявлення помилки;
- 2) ідентифікація залишків, в яких виникли помилки;
- 3) виправлення помилок.

Виявлення помилки в залишках відбувається шляхом перевірки, чи виходить величина X' за рамки робочого діапазону.

Найпростіший спосіб обчислення величини X' – це виконати перетворення із СЗК в позиційну систему числення, використовуючи КТЗ або перетворення з використанням змішаних систем числення. Ці два способи зазвичай вимагають тривалого часу обчислення та складного апаратного забезпечення.

З іншого боку, метод виявлення помилок з використанням синдрому дозволяє уникнути повної оцінки величини всіх отриманих залишків. Для виявлення помилки в залишках потрібно обчислити більше одного синдрому. При цьому обчислення синдромів також вимагає багато часу, але їх обчислення може виконуватися паралельно. Після виявлення помилки в залишку, наступним важливим і складним кроком є ідентифікація залишків, у яких відбулася помилка.

1.3.1 Алгоритми виявлення та корекції одиничних помилок

Перший алгоритм виявлення та корекції одиничних помилок був запропонований в [78]. Ввівши два надлишкові модулі p_{k+1} p_{k+2} до СЗК з модулями (p_1, p_2, \dots, p_k) , алгоритм здатний виявляти та виправляти одиничні помилки при наступних умовах [46]:

- 1) p_{k+1} є більшим за будь-який інформаційний модуль;
- 2) p_{k+2} є у два рази більший, ніж будь-який інформаційний модуль;
- 3)
$$\prod_{i=k+1}^{k+2} p_i > 2 \sum_{i=1}^k (p_i - 1) + \sum_{i=k+1}^{k+2} (m_i - 1).$$

Таким чином, підходи, які використовуються існуючими алгоритмами для виявлення та виправлення одиничних помилок, можна розділити на три групи:

- 1) обчислення синдрому;
- 2) визначення діапазону;
- 3) обчислення проєкцій (таблиці 1.1).

Серед розглянутих підходів обчислення синдрому вимагає лише однієї ітерації обчислення для виявлення та виправлення помилки. Таким чином, алгоритми, засновані на такому підході, краще піддаються апаратній реалізації. Складність обчислень визначається вибраним набором модулів. Залежно від алгоритмів, після виправлення помилки вихідний сигнал буде подано у вигляді залишку або в позиційній формі, як зазначено в таблиці 1.2.

Для додатків, які потребують виправлення помилок при обробці даних, подання сигналу у формі залишку має переваги, тому що подальші арифметичні операції в залишках можуть бути виконані безпосередньо, не витрачаючи засобів на перетворення. Подання вихідного сигналу у позиційній формі має переваги, коли потрібна корекція помилок при виконанні арифметичних операцій, для яких наступні арифметичні операції або обробка повинні виконуватися у позиційній формі. У цьому випадку витрати на зворотне перетворення можуть бути усунені.

Таблиця 1.2 – Порівняння алгоритмів виявлення та виправлення помилок в СЗК

Алгоритм	Підхід	Кількість виправлених помилок	Форма вихідного сигналу
[68]	Обчислення синдрому	Одна	Залишки
[64]	Діапазон виявлення	Одна	Ціле число
[13]	Модульні проєкції	Одна	Ціле число
[26]	Модульні проєкції	Одна	Ціле число
[42]	Обчислення синдрому	Одна	Залишки
[65]	Обчислення синдрому	Дві і більше	Залишки
[29]	Синдром	Дві і більше	Залишки
[29]	КТЗ+ проєкції	Дві і більше	Ціле число
[66]	Синдром	Дві і більше	Залишки

Отже, незважаючи на високі коригуючі можливості кодів СЗК, алгоритми виявлення та виправлення помилок на основі СЗК мають високу обчислювальну та апаратну складність [46]. Це пов'язано з ітераційними обчисленнями, необхідними для послідовного пошуку помилок у залишках. Кількість обчислень залежить від величини та кількості вибраних модулів. Оскільки при збільшенні кількості помилок, які необхідно коригувати, зростають значення модулів, відповідно, зростає й обчислювальна складність виявлення та виправлення множини помилок. Тому складність апаратної реалізації виправлення помилок є співмірна зі складністю реалізації декількох міжмодульних операцій у СЗК.

Відповідно, зменшення обчислювальної та апаратної складності методів виявлення та виправлення помилок на основі надлишкової СЗК залишається актуальною задачею.

1.3.2 Системи залишкових класів зі спеціальною системою модулів

Реалізація методів виправлення помилок з використанням коригуючих кодів у системі залишкових класів вимагає значних обчислювальних та апаратних ресурсів, що значно обмежує їх практичне використання, зокрема в безпроводних сенсорних мережах. Отже, важливою є задача підвищення ефективності реалізації методів виявлення та виправлення помилок на основі коригуючих кодів у СЗК.

Для зменшення апаратної та обчислювальної складності при виконанні перетворень з позиційної системи числення в СЗК використовують системи спеціальних модулів [50]:

- 1) $(2^n - 1, 2^n, 2^n + 1)$;
- 2) $(2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1)$;
- 3) $(2^{n-1} - 1, 2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1)$ та інші.

Для обчислення залишків числа X за спеціальними модулями типу $2^n + 1$, 2^n і $2^n - 1$ розділяємо X на три n – бітові блоки B_1, B_2, B_3 [65]:

$$B_1 = \sum_{j=2n}^{3n-1} x_j \cdot 2^{j-2n},$$

$$B_2 = \sum_{j=n}^{2n-1} x_j \cdot 2^{j-n},$$

$$B_3 = \sum_{j=0}^{n-1} x_j \cdot 2^j.$$

Тоді

$$b_1 = |B_1 - B_2 + B_3|_{2^n + 1},$$

$$b_3 = |B_1 + B_2 + B_3|_{2^n - 1}.$$

Залишком b_2 числа X за модулем 2^n є молодші n бітів числа X , відповідно, обчислення залишку зводиться до зсуву вправо розрядів числа X на n бітів, тобто $b_2 = B_2$.

Використання наведених систем модулів забезпечує зменшення апаратних затрат при реалізації суматорів, помножувачів, операції отримання залишку та виконанні інших немодульних операцій [73].

Проведені дослідження операції обчислення залишку з використанням довільної та спеціальної системи модулів показали, що використання системи спеціальних модулів забезпечує зменшення апаратних затрат в 2,8 рази та часу виконання перетворення на 29 % (таблиця 1.3).

Таблиця 1.3 – Обчислення залишків з використанням довільної та спеціальної системи модулів

Тип модулів	Кількість логічних елементів, шт	Час виконання перетворення, нс
Довільні	3766	45.614
Спеціальні	1325	32.513

Залежність апаратних затрат обчислення залишку від розрядності модулів типу $2^n - 1$, $2^n + 1$ та довільними значеннями модулів (АМ) приведені на рисунку 1.8.

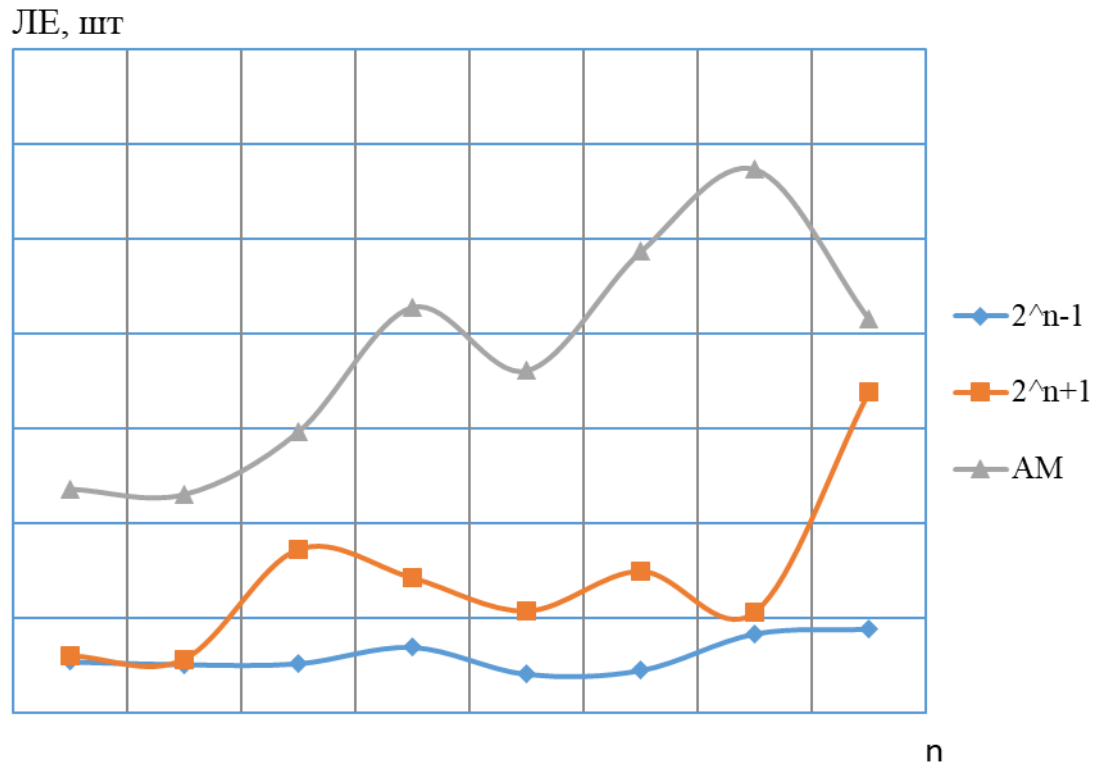


Рисунок 1.8 – Залежність апаратних затрат від розрядності модулів при обчисленні залишків

Як видно з рисунку 1.8, найменше апаратних затрат необхідно для реалізації обчислення залишку за модулем $2^n - 1$.

1.3.3 Модулярні коригуючі коди

У [87] розроблено інший принцип формування коригуючих кодів у СЗК, так звані слабо арифметичні коди СЗК, в яких значення контрольного розряду визначається:

$$\bar{b}_{k+1} = (\lambda_1 \cdot b_1 + \lambda_2 \cdot b_2 + \dots + \lambda_i \cdot b_i + \dots + \lambda_k \cdot b_k) \pmod{p_{k+1}},$$

де λ_i – коефіцієнти взаємно прості з p_{k+1} ; b_i – залишки за відповідними модулями p_i , $i = 1, \dots, k$.

Недоліком даних кодів є те, що вхідні дані мають бути подані в системі залишкових класів, що потребує додаткових затрат на перетворення даних.

У [110] розроблено модулярні коригуючі коди, які зберігають переваги слабо арифметичних кодів СЗК, проте на відміну від останніх, на вхід кодера дані подаються й обробляються в позиційній системі числення.

У модулярних коректуючих кодах пакет даних, який підлягає передачі, розбивається на k блоків однакової довжини (тетради або байти) і значення перевірного символу обчислюється [79, 115, 116]

$$x_{k+1} = |v_1 \cdot x_1 + v_2 \cdot x_2 + \dots + v_i \cdot x_i + \dots + v_k \cdot x_k|_p,$$

де v_i – коефіцієнти, взаємно прості з p ; x_i – байт даних у двійковій або десятковій системі числення.

З метою підвищення ефективності застосування модулярних коригуючих кодів у безпроводних сенсорних мережах необхідно розробити нові та удосконалити існуючі методи та алгоритми виявлення та виправлення помилок на основі модулярних коригуючих кодів, зокрема, розробити метод виправлення двох помилок з використанням одного перевірного модуля та розробити метод виправлення пакетів помилок.

Висновки до першого розділу

1. Проведено аналіз розробки та впровадження безпроводних сенсорних мереж (БСМ). Виділено переваги та невирішені задачі.
2. Проведено аналіз методів та алгоритмів, які використовуються для підвищення надійності передачі даних у безпроводних сенсорних мережах. Встановлено, що методи, які використовуються сьогодні, не забезпечують необхідну надійність передачі даних або мають високу надлишковість.

3. Проведено аналіз алгоритмів виявлення та виправлення помилок на основі надлижкової системи залишкових класів. Показано переваги використання коригуючих кодів системи залишкових класів. Вказані можливі шляхи зменшення обчислювальної складності алгоритмів кодування/декодуювання в системі залишкових класів.

4. Проведені дослідження операції обчислення залишку з використанням довільної та спеціальної системи модулів показали, що використання системи спеціальних модулів забезпечує зменшення апаратних затрат у 2,8 рази та часу виконання перетворення на 29 %.

РОЗДІЛ 2

МЕТОДИ ЗАВАДОСТІЙКОГО КОДУВАННЯ ДАНИХ В БЕЗПРОВІДНИХ
СЕНСОРНИХ МЕРЕЖАХ НА ОСНОВІ МОДУЛЯРНОЇ АРИФМЕТИКИ

2.1 Корируючі коди в системі залишкових класів зі спеціальною системою модулів

Наведені в літературі системи спеціальних модулів мають недостатню кількість членів, при заданому значенні n , для реалізації корируючих кодів у системі залишкових класів [105].

Тому в даній роботі запропоновано використання розширеної системи модулів типу: 2^n , $2^n + 1$, $2^{n+1} - 1$, $2^{n+2} - 1$, $2^{n+2} + 1$, $2^{n+3} - 1$, яка зберігає переваги спеціальної системи модулів і при цьому забезпечує необхідну їх кількість.

Методи та алгоритми виявлення та виправлення помилок з використанням корируючих кодів СЗК можна поділити на дві групи: 1) послідовне обчислення проєкцій числа за $n-1$ модулем; 2) обчислення синдрому та виправлення помилки з використанням таблиць синдрому [73].

2.1.1 Виправлення помилок на основі обчислення проєкцій числа

Виявлення та виправлення помилок на основі обчислення проєкцій числа базується на визначенні: нехай $X = (x_1, x_2, \dots, x_i, \dots, x_n)$ правильне число, тоді значення X не зміниться, якщо його представляти в системі модулів, із якої вилучено модуль p_i . Значення X_i , отримане із X без модуля p_i , називають проєкцією числа X за модулем p_i . Відповідно, якщо число $X = (x_1, x_2, \dots, x_i, \dots, x_n)$ правильне, то проєкції цього числа за всіма модулями співпадають: $X_1 = X_2 = \dots = X_i = \dots = X_n < P_k$ [106].

Алгоритм виявлення та виправлення помилок складається з наступних кроків:

1. Виявлення помилок. Обчислення позиційного представлення числа X , при цьому, якщо отримане число $X < P_k$ – помилки немає, або спотворення зазнали залишки за двома і більше модулями. В іншому випадку помилку виявлено.

2. Виправлення помилки:

2.1 Обчислення проєкцій числа X за всіма модулями p_i . Якщо проєкція $X_i < P_k$, то помилка відбулася за модулем p_i .

2.2 Виправлення помилки полягає в отриманні залишку із проєкції X_i за модулем p_i : $x_i = X_i \pmod{p_i}$.

2.1.2 Виправлення помилок на основі обчислення синдрому

Обчислення синдрому оперує з меншим діапазоном чисел за рахунок відновлення позиційного представлення числа окремо за модулем робочого та перевірного діапазонів, що відповідно має меншу апаратну складність. Розглянемо даний підхід більш детально.

Нехай $X' \equiv (x'_1, x'_2, \dots, x'_i, \dots, x'_n)$ – прийняте повідомлення представлене в непозиційній системі числення. Для виявлення помилки обчислюємо синдром s :

$$s = \left| \left| X' \right|_{P_k} - \left| X' \right|_R \right|, \quad (2.2)$$

де $\left| X' \right|_{P_k} \equiv (x'_1, x'_2, \dots, x'_k)$, $\left| X' \right|_R \equiv (x'_{k+1}, x'_{k+2})$, P_k – робочий діапазон, R – перевірений діапазон, $\left| \bullet \right|_P$ – операція отримання залишку за модулем P .

При $s = 0$ – помилка відсутня, $s \neq 0$ – наявна помилка. Оскільки будь-яка помилка призводить до різного значення синдрому, то виправлення помилки здійснюється за допомогою попередньо обчисленої таблиці синдромів для кожного модуля [105].

Для дослідження апаратної складності та швидкодії виправлення помилок (часу формування вихідних сигналів) вибрано дві системи модулів: з довільними модулями ($p_1 = 19, p_2 = 29, p_3 = 37, p_4 = 43, p_5 = 59, p_6 = 83$) і спеціальними модулями, при $n = 4$ ($p_1 = 16, p_2 = 17, p_3 = 31, p_4 = 63, p_5 = 65, p_6 = 127$). Модулі p_1, p_2, p_3, p_4 – інформаційні, p_5, p_6 – перевірочні. Використання двох перевірочних модулів дозволяє виявити та виправити помилки в будь-якому залишку за одним модулем [73].

Нехай передали повідомлення $X = 1520 = (0, 7, 1, 8, 25, 123)$, у результаті спотворення отримали повідомлення $X' = (0, 5', 1, 8, 25, 123)$ – помилка в символі за модулем p_2 .

Робочий діапазон: $P_k = p_1 \cdot p_2 \cdot p_3 \cdot p_4 = 16 \cdot 17 \cdot 31 \cdot 63 = 531216$.
 Перевірочний діапазон: $R_r = p_5 \cdot p_6 = 65 \cdot 127 = 8255$. Використовуючи попередньо знайдені ортогональні базиси, знаходимо $|X'|_{P_k} = 531216$ і $|X'|_R = 8255$. Для виявлення помилки обчислимо синдром $\delta = |531216 - 8255|_R = 3501$. Оскільки синдром не дорівнює нулю, помилка виявлена. Використовуючи таблицю синдромів (таблиця 2.1), які відповідають певному значенню помилки, знаходимо правильне значення інформаційного символу за формулою:

$$e = |x' - x|_p,$$

$$x = |x' - e|_p. \quad (2.3)$$

Таблиця 2.1 – Значення синдрому та помилки за модулем $p_2 = 17$.

№	s	e_i
0	6483	2
1	7045	3
2	4711	4
3	5273	5
4	2939	6
5	3501	7
6	1167	8
7	1729	9
8	7650	10
9	8212	11
10	5878	12
11	6440	13
12	4106	14
13	4668	15
14	2334	16
16	562	1

Для проведення експериментальних досліджень реалізовано пристрої виявлення і виправлення помилок на основі методів обчислення проекції та обчислення синдрому. Пристрої реалізовані на ПЛІС фірми Altera з використанням мови опису апаратних засобів Verilog (таблиця 2.2). Моделювання та верифікацію роботи здійснено в середовищі Quartus фірми Altera.

Таблиця 2.2 – Експериментальні результати дослідження апаратної та часової складності пристроїв виправлення помилок

Параметр	Метод обчислення проекції		Метод обчислення синдрому	
	Довільні модулі	Спеціальні модулі	Довільні модулі	Спеціальні модулі
Кількість логічних елементів, шт.	25833	19284 (26%)	49135	44398 (10%)
Затримка встановлення сигналу, мс	89,646	80,936 (18%)	68,142	61,802 (9%)

Проведені дослідження методів виправлення помилок на основі коригуючих кодів СЗК показали, що реалізація пристроїв виявлення та виправлення помилок на програмованих логічних інтегральних схемах з використанням спеціальної системи модулів забезпечує зменшення апаратних затрат приблизно на 26%, що становить 6,5 тис. логічних елементів для методу проекцій і приблизно 10 % для методу обчислення синдрому. Використання спеціальної системи модулів підвищує також швидкість роботи декодера приблизно на 18 % для методу обчислення проекцій та на 9 % для методу обчислення синдрому.

Результати досліджень підтвердили, що коригуючі коди системи залишкових класів зі спеціальною системою модулів забезпечують зменшення апаратної складності та підвищення швидкодії при реалізації кодерів/декодерів на програмованих логічних інтегральних схемах як на етапі отримання залишків, так і на етапі виявлення та виправлення помилок.

2.2 Метод формування перевірочних символів в коригуючих кодах системи залишкових класів

Суттєвим обмеженням використання коригуючих кодів СЗК є необхідність додаткового перетворення даних з двійкової системи числення в систему залишкових класів.

У роботі розроблено новий метод формування коригуючих кодів СЗК, суть якого полягає в наступному. Послідовність бітів, яка підлягає передачі, розділяється на k частин по 4 або 8 біт:

$$(a_1^1 a_2^1 \dots a_i^1 \dots a_m^1, a_1^2 a_2^2 \dots a_i^2 \dots a_m^2, \dots, a_1^j a_2^j \dots a_i^j \dots a_m^j, \dots, a_1^k a_2^k \dots a_i^k \dots a_m^k), \quad (2.4)$$

де a^i – розряд даних у двійковому коді, $m = 4, 8$.

Кожній частині двійкового коду ставляться у відповідність прості числа (модулі) p_i ($p_1 < p_2 < \dots < p_i < \dots < p_n$) з яких перші k модулів інформаційні, n – загальна кількість модулів, $r = n - k$ – перевірочні модулі. Значення модулів вибираємо з умови $p_i > 2^m$. При цьому перші k модулів визначають робочий

$$\text{діапазон } P_k = \prod_{i=1}^k p_i, \text{ повний діапазон дорівнює } P = \prod_{i=1}^n p_i.$$

Оскільки значення тетрад або байтів у позиційному представленні менші ніж відповідні модулі p_i , то їх можна вважати залишками. У результаті вказаного перетворення повідомлення набуде вигляду:

$$(x_1, x_2, \dots, x_i, \dots, x_k), \quad (2.5)$$

де x_i – частини повідомлення, які одночасно є залишками по вибраних модулях

$$p_i, \quad x_i = \sum_{i=1}^m a_i \cdot 2^i.$$

Для обчислення перевірочних символів повідомлення (2.5) перетворимо в позиційну систему числення

$$X = \left(\sum_{i=1}^k x_i \cdot M_i \cdot \delta_i \right) \bmod P_K,$$

де $M_i = \frac{P_K}{p_i}$, $\delta_i = M_i^{-1} \bmod p_i$.

Перевірочні символи обчислюються за формулою [108]:

$$x_{k+i} = X \bmod p_{k+i}, \quad i = \overline{1, (n-k)},$$

де X – повідомлення в позиційній системі числення.

У результаті кодове слово складається з інформаційних і перевірочних символів та має вигляд:

$$(x_1, x_2, \dots, x_i, \dots, x_k, x_{k+1}, \dots, x_n).$$

Виявлення помилок. Якщо в прийнятому повідомленні $X' = (x'_1, x'_2, \dots, x'_i, \dots, x'_k, x_{k+1}, \dots, x'_n)$ відбулась помилка, то його позиційне представлення вийде за межі робочого діапазону, тобто $X' > P_k$ [102].

Нехай $X = 1010011101011001$ – повідомлення, яке необхідно передати. Розділимо дане повідомлення X на тетради: $x_1 = 1010$, $x_2 = 0111$, $x_3 = 0101$, $x_4 = 1001$. Модулі виберемо згідно з умовою $p_i > 2^4$: $p_1 = 17$, $p_2 = 19$, $p_3 = 23$; $p_4 = 29$ – інформаційні, $p_5 = 31$ – перевірочний модуль. Робочий діапазон становить $P_K = 17 \cdot 19 \cdot 23 \cdot 29 = 215441$. Загальний діапазон $P = P_K \cdot p_5 = 215441 \cdot 31 = 6678671$. Оскільки значення x_1 , x_2 , x_3 , x_4 в десятковій системі числення менші ніж відповідні модулі, то їх можна вважати залишками за даними модулями. Переведемо повідомлення $X = (x_1, x_2, x_3, x_4)$ в

десяткову систему числення. Для цього обчислимо ортогональні базиси:

$$M_1 = \frac{P_K}{P_1} = 12673, \quad M_2 = 11339, \quad M_3 = 9367, \quad M_4 = 7429. \quad \text{Обернені числа до}$$

$$M_1 - M_4 \quad \text{рівні} \quad \delta_1 = 15, \quad \delta_2 = 14, \quad \delta_3 = 4, \quad \delta_4 = 6. \quad \text{Отже,}$$

$$X = \sum_{i=1}^k (x_i \cdot M_i \cdot \delta_i) \bmod P_K = 153622.$$

Перевірочний символ обчислюється за формулою
 $x_5 = X \bmod p_5 = 153622 \bmod 31 = 17.$

Таким чином, повідомлення після кодування має вигляд:
 $X' = (10, 7, 5, 9, 17)$ або $X' = (1010, 0111, 0101, 1001, 10001).$

Результати експериментальних досліджень. У даному розділі представлені результати експериментальних досліджень апаратних затрат (кількість логічних елементів) та часу обчислення перевірочних символів при різній розрядності вхідних даних для коригуючих кодів СЗК та для запропонованого методу формування коригуючих кодів СЗК. Експериментальні дослідження проводилися з використанням програмного забезпечення Quartus фірми Intel (Altera). Відомий та запропонований метод формування коригуючих кодів СЗК описані на мові програмування апаратних засобів Verilog-HDL та синтезовані в мікросхемах Cyclone IV.

Експериментальні дослідження проведені при наступних даних:

- розрядність вхідних даних від 16 до 48 біт;
- кількість інформаційних модулів - 4;
- кількість перевірочних модулів - 1, 2;
- значення модулів залежить від розрядності повідомлення.

Проведено дослідження запропонованого методу формування перевірочних символів у коригуючих кодах СЗК (метод 2) з відомими коригуючими кодами СЗК (метод 1) [76].

На рисунку 2.1, 2.2 приведено результати експериментальних досліджень апаратних затрат (кількість логічних елементів) та часу обчислення

перевірочних символів при різній розрядності вхідних даних для коригуючих кодів СЗК та для запропонованого методу з одним перевірочним модулем.

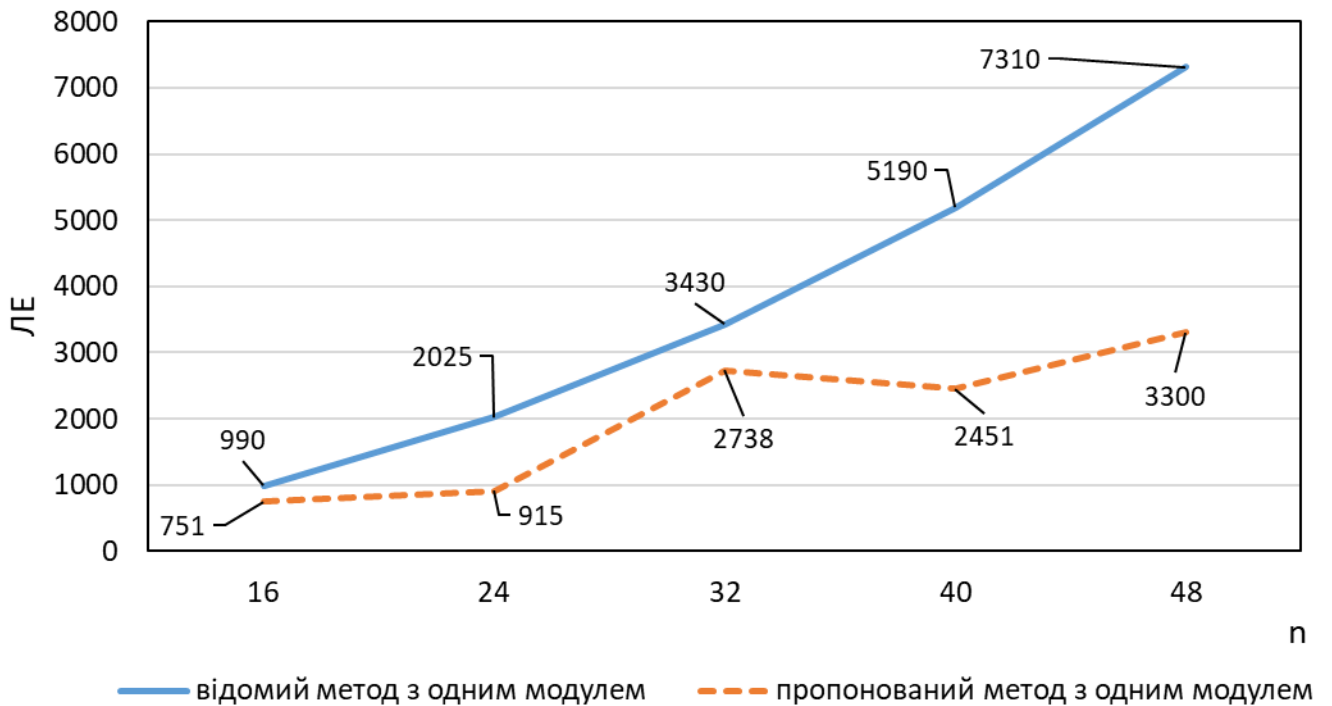


Рисунок 2.1 – Залежність апаратних затрат від розрядності вхідних даних при одному перевірочному модулі для методу 1 і методу 2

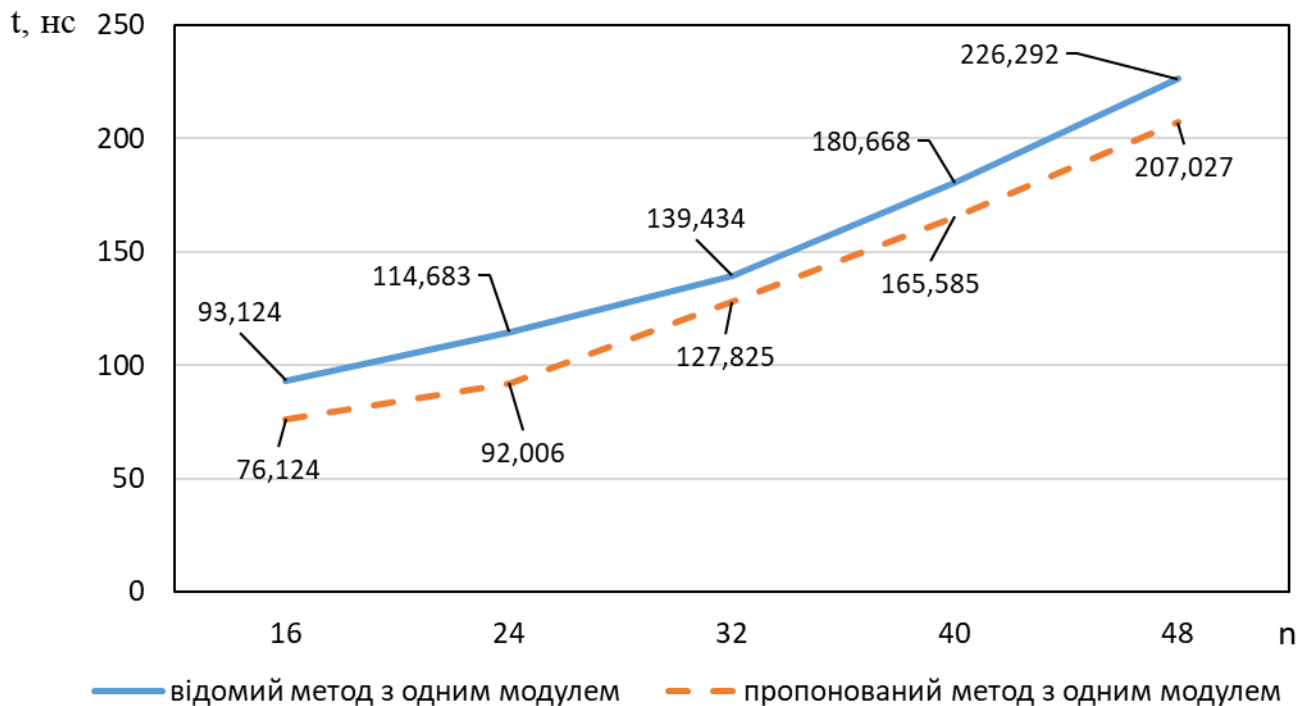


Рисунок 2.2 – Залежність часу обчислення перевірочних символів від розрядності вхідних даних при одному перевірочному модулі для методу 1 і 2

Як видно з рисунку 2.1 запропонований метод забезпечує зменшення апаратних затрат в середньому на 35% залежно від розрядності вхідних даних. Часові затрати зменшуються в середньому на 5% (рисунок 2.2).

На рисунку 2.3, 2.4 приведено результати експериментальних досліджень апаратних затрат (кількість логічних елементів) та часу обчислення перевірочних символів при різній розрядності вхідних даних, для коригуючих кодів СЗК та для запропонованого методу з двома перевірочними модулями.

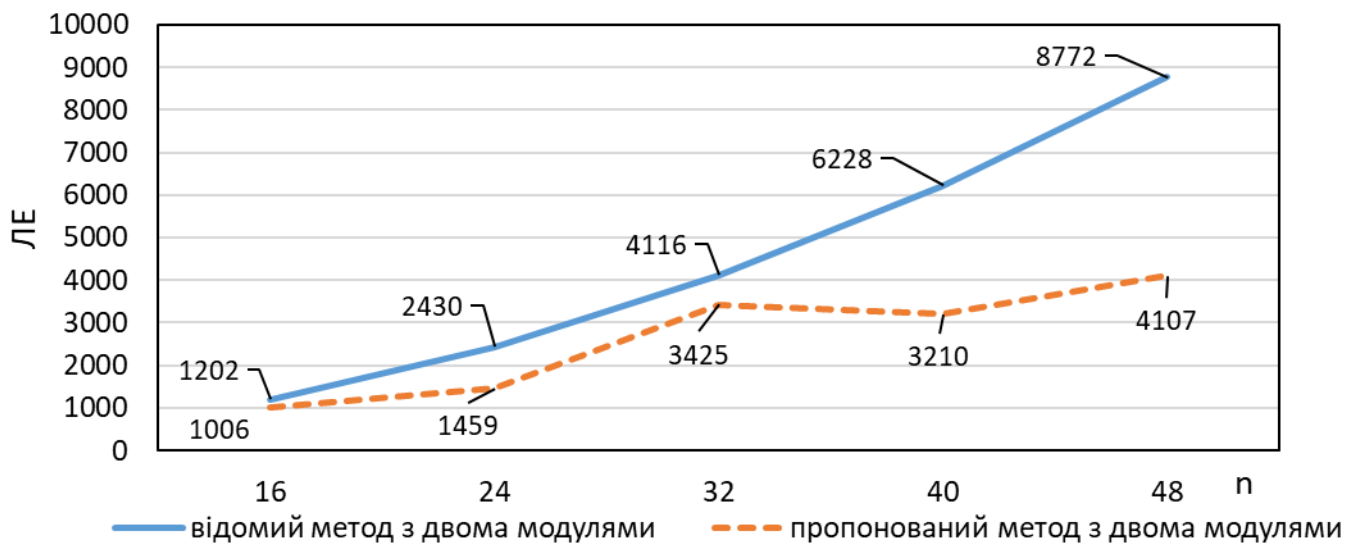


Рисунок 2.3 – Залежність апаратних затрат від розрядності вхідних даних при двох перевірочних модулях для методу 1 і 2

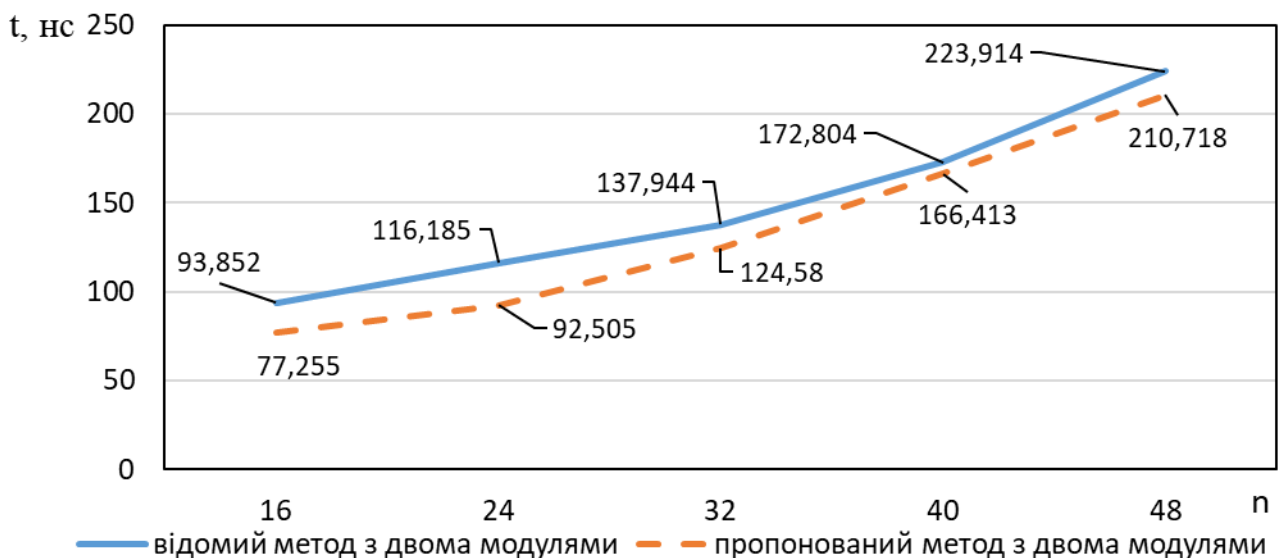


Рисунок 2.4 – Залежність часу обчислення перевірочних символів від розрядності вхідних даних при двох перевірочних модулях для методу 1 і 2.

Як видно з рисунку 2.3 запропонований метод забезпечує зменшення апаратних затрат в середньому на 29% і залежить від розрядності вхідних даних. При цьому часові затрати зменшуються в середньому на 3-5% (рисунок 2.4).

Також проведено дослідження апаратних затрат (рисунок 2.5) та швидкодії обчислення (рисунок 2.6) при поділі вхідних даних на 8 біт. При цьому значення всіх модулів є більші за 256.

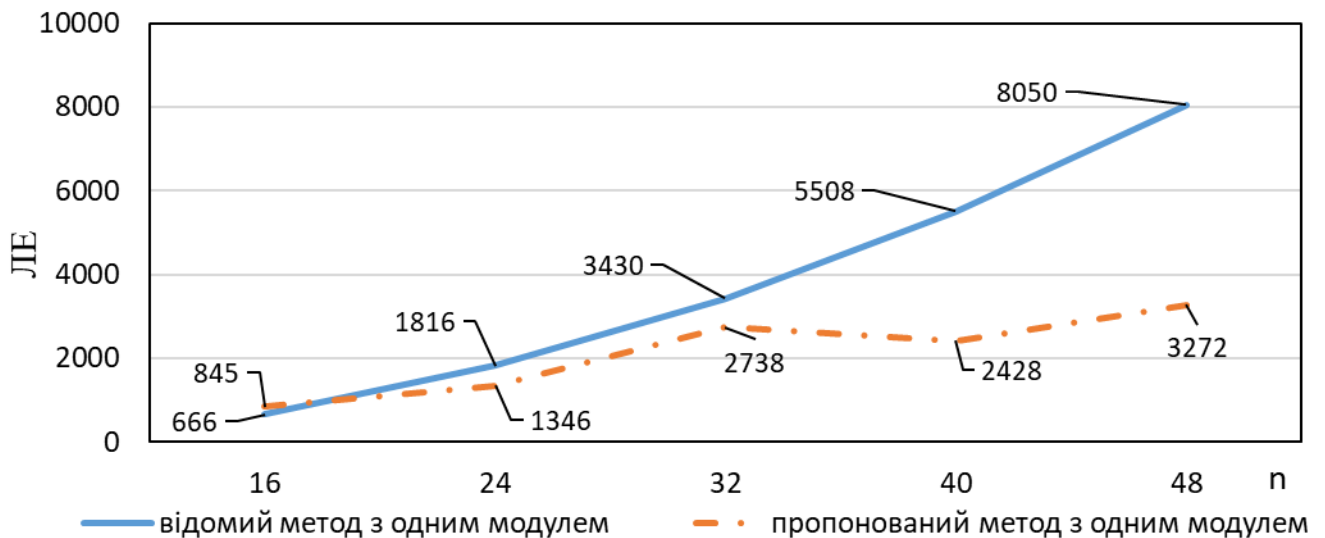


Рисунок 2.5 – Залежність апаратних затрат від розрядності вхідних даних при одному перевірочному модулі для методу 1 і 2 (розрядність блоку 8 біт).

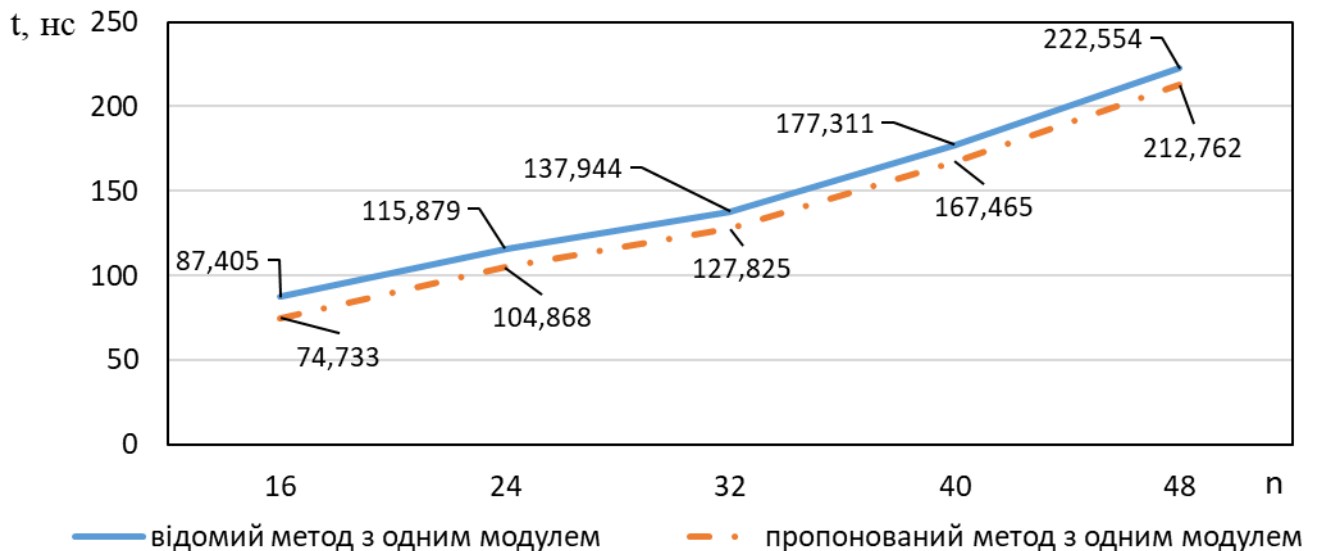


Рисунок 2.6 – Залежність часу обчислення перевірочних символів від розрядності вхідних даних при одному перевірочному модулі для методу 1 і 2 (розрядність блоку 8 біт).

Як видно з рисунку 2.5 запропонований метод забезпечує зменшення апаратних затрат в середньому на 23% і залежить від розрядності вхідних даних (розрядність блоку 8 біт). При цьому часові затрати зменшуються в середньому на 5% (рисунок 2.6).

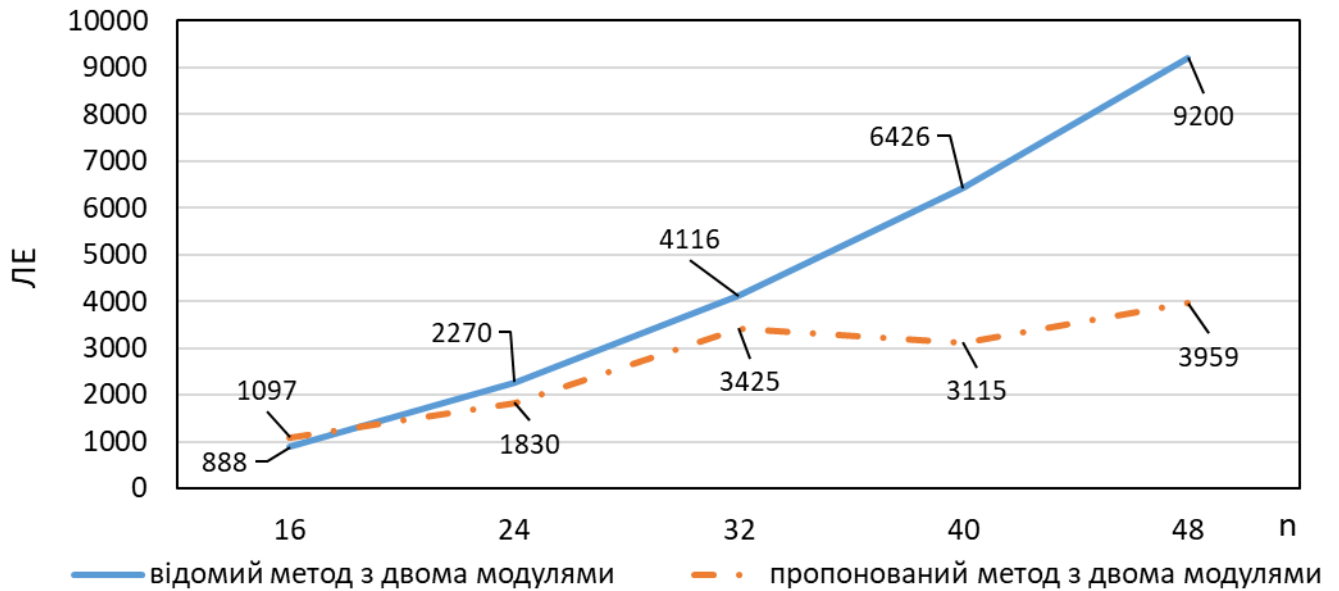


Рисунок 2.7 – Залежність апаратних затрат від розрядності вхідних даних при двох перевірючих модулях для методу 1 і 2 (розрядність блоку 8 біт).

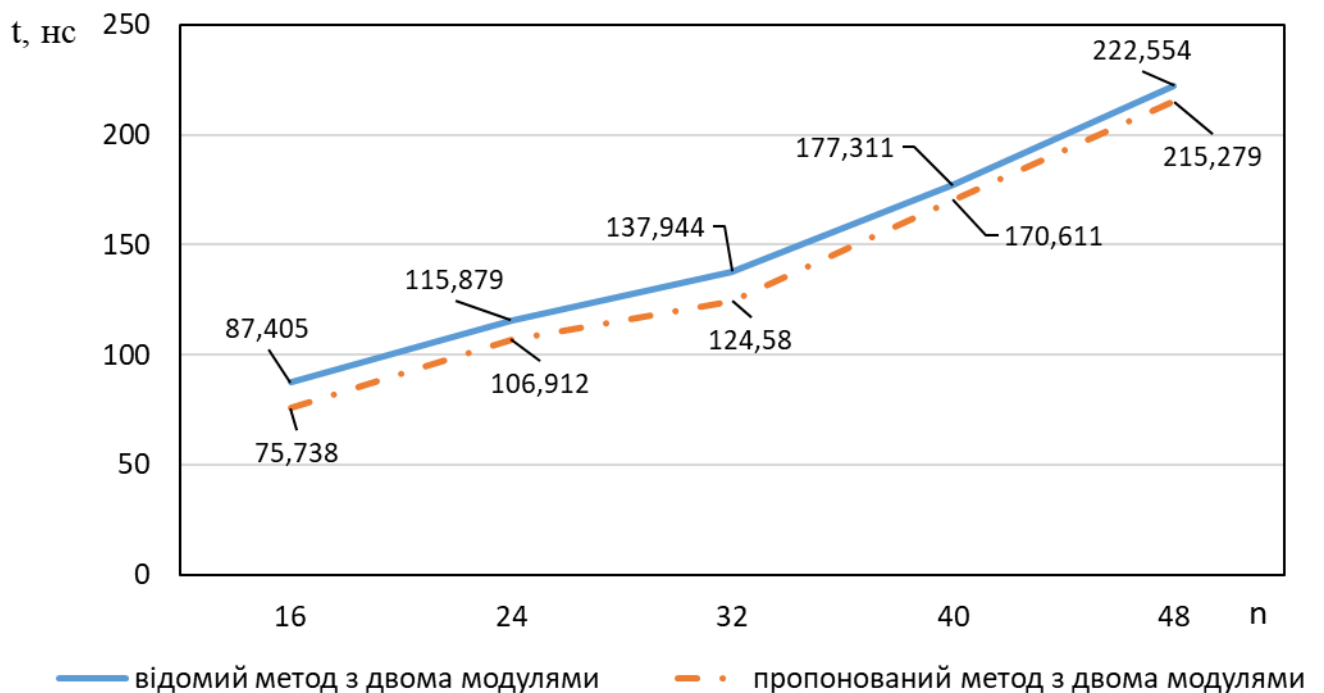


Рисунок 2.8 – Залежність часу обчислення перевірючих символів від розрядності вхідних даних при двох перевірючих модулях для методу 1 і 2 (розрядність блоку 8 біт).

Як видно з рисунку 2.7 запропонований метод забезпечує зменшення апаратних затрат в середньому на 20% і залежить від розрядності вхідних даних (розрядність блоку 8 біт). При цьому часові затрати зменшуються в середньому на 5 % (рисунок 2.8).

Запропонований метод формування коригуючих кодів СЗК забезпечує зменшення апаратних затрат на 20% залежно від розрядності повідомлення та підвищення швидкодії за рахунок відсутності процедури перетворення вхідного повідомлення в СЗК.

Таким чином, розроблений метод формування перевірочних символів значно розширить область застосування коригуючих кодів СЗК за рахунок обробки повідомлень, які представлені в позиційних системах числення.

2.3 Метод виправлення помилок в двох інформаційних символах на основі модулярних коригуючих кодів з одним перевірочним символом

Розроблено метод виправлення помилок в двох інформаційних символах з використанням одного перевірочного символу.

Значення контрольного символу в модулярних коригуючих кодах обчислюється за формулою [71, 109, 112]

$$x_{k+1} = |(v_1 \cdot x_1 + v_2 \cdot x_2 + \dots + v_i \cdot x_i + \dots + v_k \cdot x_k)|_P, \quad (2.6)$$

де x_i – інформаційні символи, v_i – коефіцієнти взаємно прості з P , $|\bullet|_P$ – операція отримання залишку по модулю P .

Декодер по прийнятих даних $(x'_1, x'_2, \dots, x'_i, \dots, x'_k)$ обчислює значення контрольного символу:

$$x'_{k+1} = \left| (v_1 \cdot x_1 + v_2 \cdot x_2 + \dots + v_i \cdot x'_i + \dots + v_k \cdot x_k) \right|_P. \quad (2.7)$$

Для виявлення помилки обчислимо синдром δ , який є різницею між перевірочним символом отриманим і перевірочним символом обчисленим на приймальній стороні (у декодері):

$$\delta = \left| x'_{k+1} - x_{k+1} \right|_P, \quad (2.8)$$

рівняння (2.6) можна записати у вигляді

$$\delta = \left| v_1 \cdot (x'_1 - x_1) + v_2 \cdot (x'_2 - x_2) + \dots + v_i \cdot (x'_i - x_i) + \dots + v_k \cdot (x'_k - x_k) \right|_P, \quad (2.9)$$

якщо синдром дорівнює нулю $\delta = 0$, то помилки немає, оскільки при відсутності помилки $x'_i = x_i$, отже $x'_{k+1} = x_{k+1}$, якщо $\delta \neq 0$ – наявна помилка, відповідно $x'_i \neq x_i$ і, як наслідок – $x'_{k+1} \neq x_{k+1}$.

Для виправлення помилок в одному інформаційному символі необхідно і достатньо, щоб значення синдрому δ було унікальне для всіх можливих варіантів помилки. Для виконання даної умови потрібно, щоб виконувалась умова $P > 2 \cdot n \cdot (2^m - 1)$, де n – кількість інформаційних символів, m – розрядність інформаційних символів.

Розглянемо можливість виправлення помилок у двох інформаційних символах. Припустимо, що помилки відбулися в двох символах, оскільки при відсутності помилки $x'_i - x_i = 0$, то рівняння (2.9) набуде вигляду:

$$\left| v_i \cdot (x'_i - x_i) + v_{i+1} \cdot (x'_{i+1} - x_{i+1}) \right|_P = \delta, \quad (2.10)$$

виконавши перетворення отримаємо

$$\left| v_i \cdot x_i + v_{i+1} \cdot x_{i+1} \right|_P = \left| v_i \cdot x'_i + v_{i+1} \cdot x'_{i+1} - \delta \right|_P. \quad (2.11)$$

Позначимо праву частину рівняння (2.11) через c , отримаємо діофантове рівняння з двома невідомими:

$$v_i \cdot x_i + v_{i+1} \cdot x_{i+1} = |c|_P. \quad (2.12)$$

За допомогою розширеного алгоритму Евкліда знаходимо один із розв'язків рівняння (2.12). Розширений алгоритм Евкліда по заданих коефіцієнтах v_i , v_{i+1} знаходить їх найбільший спільний дільник $g = \gcd(v_i, v_{i+1})$, а також такі коефіцієнти x_g , x_{g+1} , що

$$v_i \cdot x_g + v_{i+1} \cdot x_{g+1} = g.$$

Оскільки коефіцієнти v_i , v_{i+1} – взаємно прості числа, то $g = \gcd(v_i, v_{i+1}) = 1$, отже c ділиться на g , відповідно діофантове рівняння (2.12) має розв'язок, і одним із таких розв'язків є числа:

$$\begin{cases} x_{0i} = x_g \cdot c \\ x_{0i+1} = x_{g+1} \cdot c \end{cases}$$

Усі розв'язки рівняння (2.12) обчислимо за формулою:

$$\begin{cases} x_i = |x_{0i} - k \cdot v_{i+1}|_P, \\ x_{i+1} = |x_{0i+1} + k \cdot v_i|_P, \end{cases} \quad \text{при } k \in P. \quad (2.13)$$

Враховуючи обмеження, які накладаються на x_i , зокрема $0 \leq x_i < 2^m$, тільки один з розв'язків буде відповідати діапазону представлення даних.

Таким чином, знайдені розв'язки рівняння (2.13) будуть правильними значеннями інформаційних символів. Для виявлення помилок необхідно перевірити всі пари інформаційних символів і при наявності помилки виправити за наведеним вище методом.

Алгоритм виявлення та виправлення помилок складається з наступних кроків [72]:

1. Обчислення перевірного символу x'_{k+1} по прийнятих даних.
2. Обчислення за формулою (2.8) синдрому δ .
3. Якщо $\delta = 0$ – помилки не має.
4. Інакше обчислення виразу $f_j = |v_i * e_j|_P$.
5. Порівняння значення синдрому δ і f_j .
6. Якщо $\delta = f_j$ – помилка виявлена.
7. Виправлення помилки за формулою $x_i = |x'_i \pm e_j|$.
8. Якщо $\delta \neq f_j$ помилка наявна в більше як одному символі.
9. Припускаємо, що помилки є в символах $x_i, x_j, i \neq j$.
10. За розширеним алгоритмом Евкліда знаходимо один з розв'язків рівняння (2.10).
11. За формулою (2.13) знаходимо всі розв'язки рівняння (2.12).
12. Перевіряємо, чи є серед множини розв'язків такі, що знаходяться в діапазоні подання інформаційних символів, якщо так, то помилки виявлені, а знайдені розв'язки є правильними значеннями інформаційних символів.
13. Інакше вибираємо наступну пару символів x_i, x_j .
14. Повторювати пункти 10-13 доти поки не будуть перевірені всі комбінації інформаційних символів.

Розглянемо коригуючий код, який складається з восьми інформаційних і одного перевірного символу, та забезпечує виправлення помилок в будь-яких

двох інформаційних символах. Розрядність інформаційних символів 4 біти. Кортеж даних, які необхідно передати, має вигляд: $X = (5, 8, 10, 3, 7, 14, 12, 1)$.

Вибираємо перевірочний модуль $P = 1021$ та взаємно прості коефіцієнти для обчислення перевірочного символу: $v_1 = 13$, $v_2 = 17$, $v_3 = 19$, $v_4 = 23$, $v_5 = 29$, $v_6 = 31$, $v_7 = 37$, $v_8 = 43$.

Значення перевірочного символу знаходимо за формулою (2.6):

$$x_{k+1} = |13 \cdot 5 + 17 \cdot 8 + 19 \cdot 10 + 23 \cdot 3 + 29 \cdot 7 + 31 \cdot 14 + 37 \cdot 12 + 43 \cdot 1|_{1021} = 563.$$

У результаті спотворення даних отримали: $X = (5, 4, 6, 3, 7, 14, 12, 1)$, тобто спотворення зазнали інформаційні символи в другому і третьому розряді. Обчислимо перевірочний символ за прийнятими даними за формулою (2.7):

$$x'_{k+1} = |13 \cdot 5 + 17 \cdot 4 + 19 \cdot 6 + 23 \cdot 3 + 29 \cdot 7 + 31 \cdot 14 + 37 \cdot 12 + 43 \cdot 1|_{1021} = 419.$$

Обчислимо синдром:

$$\delta = |x'_{n+1} - x_{n+1}|_P = |419 - 563|_{1021} = 877.$$

Оскільки синдром δ не дорівнює нулю, це означає наявність помилки.

Припустимо, що помилка в інформаційних символах x_2 і x_3 . Підставивши числові значення, рівняння (2.14) запишемо у вигляді:

$$|17 \cdot (4 - x_2) + 19 \cdot (6 - x_3)|_{1021} = 877,$$

провівши обчислення, отримаємо:

$$|17 \cdot x_2 + 19 \cdot x_3|_{1021} = 326. \quad (2.14)$$

За розширеним алгоритмом Евкліда знаходимо коефіцієнти $x_{g2} = 9$, $x_{g3} = -8$, і значення $x_{02} = 892$, $x_{03} = 455$.

Підставивши значення x_{02} , x_{03} у формулу (2.13), знаходимо всі розв'язки рівняння (2.14):

$$\begin{cases} x_2 = |892 - k \cdot 19|_P, \\ x_3 = |455 + k \cdot 17|_P, \end{cases} \quad \text{при } k \in P.$$

Множина розв'язків рівняння (2.14) приведена на рисунку 2.9 (крива *Error:x2, x3*), з якого видно, що тільки один розв'язок ($x_2 = 8$, $x_3 = 10$) відповідає діапазону представлення інформаційних символів, які і відповідають правильним значенням інформаційних символів. Якщо тільки в одному з двох вибраних символів (x_2, x_3) є помилка, то серед множини розв'язків не буде правильного, тобто такого, що відповідає діапазону подання інформаційних символів (рисунок 1, крива *Error:x2,x4*).

Швидкість коду обчислюється за формулою $R = \frac{k}{n}$, де k розрядність даних, n – загальна довжина коду. Для розглянутого прикладу: $k = 32$ біт (вісім чотирьох розрядних символів), $n = 32 + \lceil \log_2 1021 \rceil = 42$, $R = 0.76$. При використанні двох перевірочних символів $n = 32 + 2 \cdot \lceil \log_2 1021 \rceil = 52$, $R = 0.62$.

Отже, використання розробленого методу дозволяє приблизно на 20% підвищити швидкість коду.

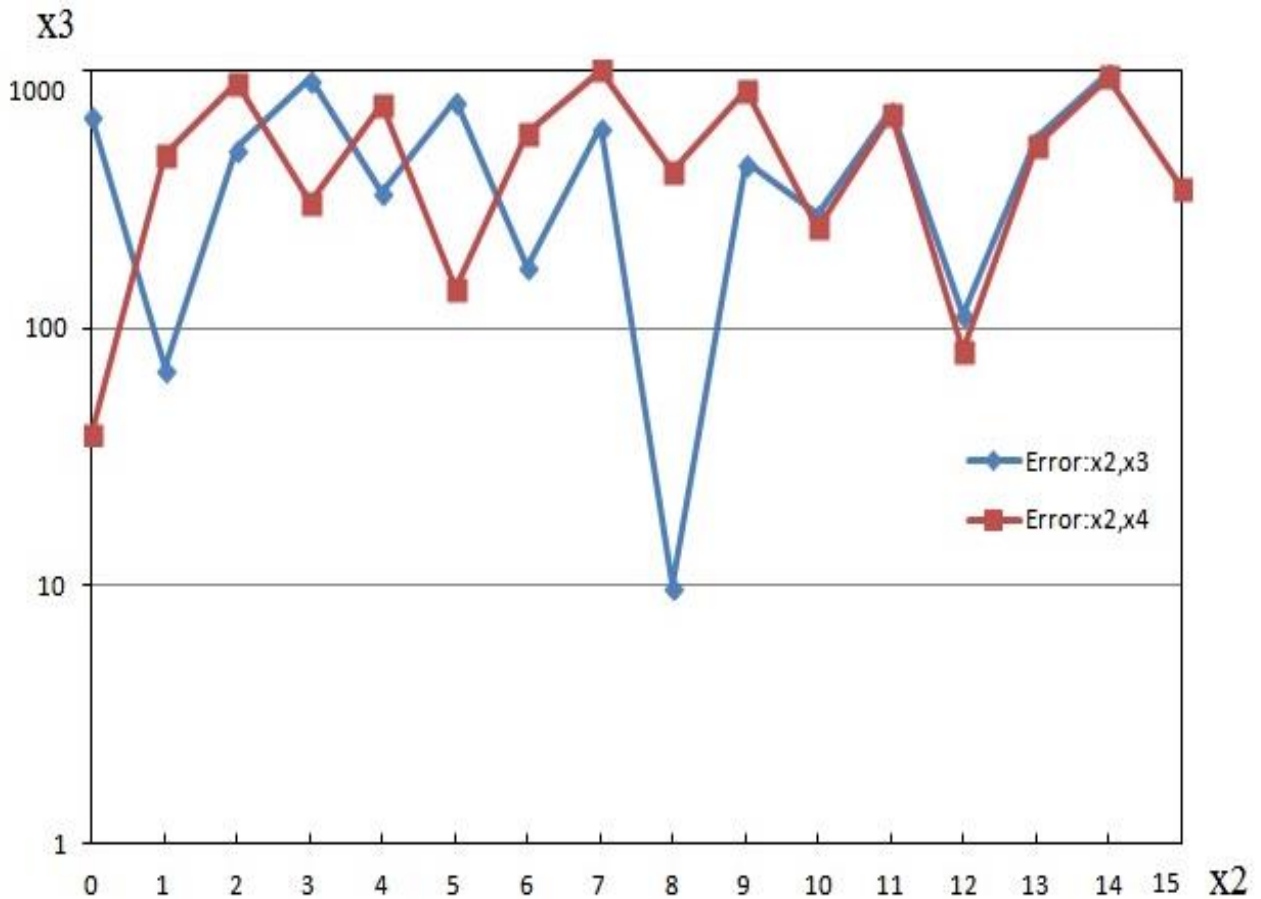


Рисунок 2.9 – Множина розв’язків діофантового рівняння (2.14)

Розроблений метод забезпечує виправлення помилок в двох інформаційних символах з використанням одного перевірного символу і тим самим дозволяє збільшити швидкість коду приблизно на 20%, відповідно – зменшити надлишковість модулярного коригуючого коду.

2.4 Метод виправлення пакетів помилок на основі модулярного коригуючого коду

Модулярні коригуючі коди, розроблені в [113, 60], належать до символічних кодів, характеризуються низькою надлишковістю та дозволяють ефективно виявляти і виправляти одиничні символічні помилки. У [107] показана можливість побудови двовимірних коригуючих кодів на основі

модулярної арифметики, які дозволяють виявляти та виправляти помилки в двох і більше інформаційних символах. Проте запропонований метод виправлення помилок у двох символах на основі розширеного алгоритму Евкліда має високу часову складність, яка залежить від величини модуля P і зростає із збільшенням кількості та розрядності інформаційних символів, оскільки потребує знаходження всіх розв'язків діофантового рівняння. Також необхідними є додаткові дослідження алгоритмів виправлення пакетів помилок. Отже, актуальною науковою задачею є розробки багатовимірного (ітеративного) методу кодування коду на основі модулярного коригуючого коду та алгоритмів виправлення пакетів помилок.

Двовимірним кодом $\gamma = \gamma_1 \times \gamma_2$ двох кодів γ_1 і γ_2 називається код, кодовими словами якого є всі двовимірні таблиці із рядками, які є словами коду γ_1 , і стовпцями, які є словами коду γ_2 .

В якості кодів γ_1 і γ_2 для побудови двовимірного коду вибрано модулярний коригуючий код [74, 75]. При цьому коди можуть бути, як однакові $\gamma_1 = \gamma_2$, так і різні $\gamma_1 \neq \gamma_2$.

У модулярному коригуючому коді перевірочні символи обчислюються:

$$f_{jk}(x) = \left(\sum_{i=1}^k x_{ij} v_i \right) \text{mod } P \quad (2.15)$$

$$f_{li}(x) = \left(\sum_{j=1}^l x_{ji} v_j \right) \text{mod } P. \quad (2.16)$$

де x_{ji} – інформаційні символи, v_i, v_j – коефіцієнти (прості числа), P – модуль (просте число).

Формування двовимірного модулярного коду наведено в таблиці 2.3. Спочатку кодом γ_1 кодується кожний із l рядків а потім кодом γ_2 кожна із k стовпців.

Таблиця 2.3 – Формування двовимірного модулярного коду

	Інформаційні символи	Перевірочні символи по рядках
	$x_{11} \quad x_{12} \quad \dots \quad x_{1i} \quad \dots \quad x_{1k}$ $x_{21} \quad x_{22} \quad \dots \quad x_{2i} \quad \dots \quad x_{2k}$ $\dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots$ $x_{j1} \quad x_{j2} \quad \dots \quad x_{ji} \quad \dots \quad x_{jk}$ $\dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots$ $x_{i1} \quad x_{i2} \quad \dots \quad x_{ii} \quad \dots \quad x_{ik}$	$f_{1k}(x)$ $f_{2k}(x)$ \dots $f_{jk}(x)$ \dots $f_{ik}(x)$
Перевірочні символи по стовпцях	$f_{11}(x) \quad f_{12}(x) \quad \dots \quad f_{1i}(x) \quad \dots \quad f_{1k}(x)$	

Процес кодування полягає в наступному (рисунок 2.10): дані по k символам поступають у внутрішній кодер γ_1 , де здійснюється обчислення перевірочних символів по рядках за формулою (2.15) (таблиця 2.3). З виходу внутрішнього кодера γ_1 кодові слова поступають у буфер. З буфера кодові слова зчитуються по стовпцях і поступають у зовнішній кодер γ_2 , де відбувається обчислення перевірочних символів по стовпцях за формулою (2.16).

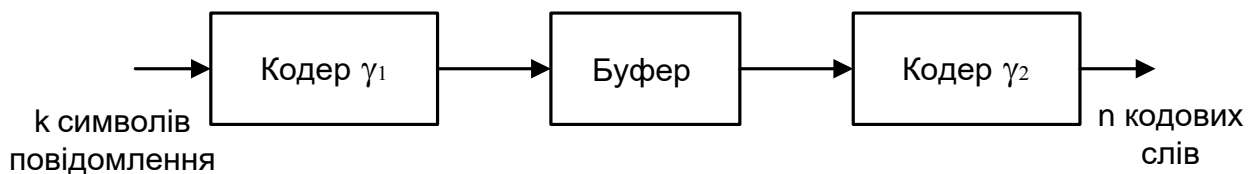


Рисунок 2.10 – Структурна схема кодера

Виявлення помилок. Виявлення помилок відбувається на основі аналізу синдрому, який обчислюється як різниця перевірочних символів, обчислених в кодері та декодері.

Обчислення синдрому по рядках:

$$\delta_i = \begin{pmatrix} (f'_{1k}(x) - f_{1k}(x)) \bmod P \\ (f'_{2k}(x) - f_{2k}(x)) \bmod P \\ \dots \dots \dots \\ (f'_{jk}(x) - f_{jk}(x)) \bmod P \\ \dots \dots \dots \\ (f'_{ik}(x) - f_{ik}(x)) \bmod P \end{pmatrix}.$$

Обчислення синдрому по стовпцях:

$$\delta_j = \begin{pmatrix} (f'_{i1}(x) - f_{i1}(x)) \bmod P \\ (f'_{i2}(x) - f_{i2}(x)) \bmod P \\ \dots \dots \dots \\ (f'_{ii}(x) - f_{ii}(x)) \bmod P \\ \dots \dots \dots \\ (f'_{ik}(x) - f_{ik}(x)) \bmod P \end{pmatrix},$$

$f'_{jk}(x), f'_{li}(x)$ – перевірочні символи обчислені в декодері по прийнятих даних.

При $\delta_i = 0, \delta_j = 0$ – помилки немає.

Якщо $\delta_i \neq 0, \delta_j \neq 0$ – наявна помилка.

Позиція спотвореного символу визначається по значенню синдромів, які на перетині рядка і стовпця не дорівнюють нулю.

Для виправлення помилки в одному символі необхідно розв'язати рівняння:

$$v_i(x'_i - x_i) = \delta_i \pmod{P}. \quad (2.17)$$

Якщо синдроми $\delta_i \neq 0, \delta_j \neq 0$ не дорівнюють нулю у двох і більше рядках або стовпцях, це означає, що помилки відбулися в двох і більше символах. При цьому, якщо синдром $\delta_j \neq 0$ в двох стовпцях – це означає, що

спотворені два символи в рядку. Позиції ненульових символів відповідають ненульовим значенням синдрому у стовпці.

Виправлення помилок у двох інформаційних символах. Аналіз синдрому по рядках і стовпцях дозволяє виявити позиції спотворених символів. Для їх виправлення необхідно розв'язати рівняння:

$$\begin{aligned} v_i(x'_i - x_i) + v_t(x'_t - x_t) &= \delta_{ij}(\text{mod } P); \\ v_i x'_i - v_i x_i + v_t x'_t - v_t x_t &= \delta_{ij}(\text{mod } P); \\ -v_i x_i - v_t x_t &= (\delta_{ij} - v_i x'_i - v_t x'_t)(\text{mod } P); \\ v_i x_i + v_t x_t &= (v_i x'_i + v_t x'_t - \delta_{ij})(\text{mod } P), \end{aligned} \quad (2.18)$$

де i, t – позиції спотворених символів.

Замінімо праву частину на c :

$$v_i x_i + v_t x_t = c(\text{mod } P). \quad (2.19)$$

Відомо, якщо $\text{gcd}(v_i, v_t) = 1$, тобто v_i і v_t взаємно прості числа, то рівняння (2.19) має розв'язок у цілих числах.

Розв'яжемо рівняння (2.19) відносно одного з невідомих. Виразимо x_i через x_t :

$$x_i = \left(\frac{c - v_t x_t}{v_i} \right) (\text{mod } P). \quad (2.20)$$

Оскільки $0 \leq x_i < 2^m$ і $0 \leq x_t < 2^m$, то підставивши в рівняння (2.20) замість x_t значення від 0 до 2^m , знаходимо множину розв'язків x_i , з яких тільки один із знайдених розв'язків буде цілим числом.

При мінімальній кодовій відстані кодів γ_1 і γ_2 рівній d_1 і d_2 відповідно то мінімальна кодова відстань двовимірного коду γ дорівнює $d_1 \times d_2$. Звідси

висновок, якщо коди γ_1 і γ_2 виправляють t_1 і t_2 помилки відповідно, то двовимірний код виправляє $2t_1t_2 + t_1 + t_2 + 1$ помилок.

Враховуючи, що модулярний код може виправляти дві помилки, якщо відомо їх розташування, то максимальна довжина пакету помилок складає $b = 3k - 2$ символи, при розміщенні спотворених символів у двох рядках і одному стовпці, або в двох стовпцях і одному рядку (рисунки 2.11).

*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
	*						
	*						
	*						
	*						
	*						
	*						

Рисунок 2.11 – Приклад розміщення пакету помилок максимальної довжини

Проте даний код не зможе виправити пакет помилок з шести символів, які розміщені на перетині трьох рядків і трьох стовпців (рисунки 2.12).

	*	*	*				
	*	*	*				
	*	*	*				

Рисунок 2.12 – Приклад розміщення пакету помилок

Розроблено двовимірний код на основі модулярних коригуючих кодів, який забезпечує ефективне виправлення пакетів помилок максимальної довжини $b = 3k - 2$ символи за умови, що спотворені символи розміщені в двох рядках і одному стовпці або в двох стовпцях і одному рядку.

Висновки до другого розділу

1. Удосконалено коригуючі коди в системі залишкових класів за рахунок вибору спеціальної системи модулів. Реалізація кодерів/декодерів на програмованих логічних інтегральних схемах з використанням спеціальної системи модулів забезпечує зменшення апаратних затрат приблизно на 26%, що становить 6,5 тис. логічних елементів для методу проєкцій та приблизно 10% для методу обчислення синдрому.

2. Запропоновано метод формування перевірочних символів у коригуючих кодах системи залишкових класів, який забезпечує зменшення апаратних затрат у середньому на 20% і залежить від розрядності повідомлення та підвищення швидкодії за рахунок відсутності процедури перетворення повідомлення з позиційної системи в СЗК і зворотного перетворення.

3. Розроблено метод виправлення помилок на основі модулярних коригуючих кодів, який забезпечує виправлення помилок у двох інформаційних символах з використанням одного перевірочного символу, і тим самим дозволяє збільшити швидкість коду приблизно на 20%, відповідно зменшити надлишковість модулярного коригуючого коду.

4. Отримав подальший розвиток двовимірний метод виправлення помилок на основі модулярних коригуючих кодів, який забезпечує ефективне виправлення пакетів помилок, характеризується високою коригуючою здатністю та низькою складністю реалізації алгоритму декодування і, відповідно, може бути застосований для підвищення надійності передачі даних у безпроводних сенсорних мережах.

РОЗДІЛ 3

АЛГОРИТМИ ЗАВАДОСТІЙКОГО КОДУВАННЯ ДАНИХ НА ОСНОВІ МОДУЛЯРНИХ КОРИГУЮЧИХ КОДІВ

3.1 Алгоритм та програмна реалізація пошуку коефіцієнтів для побудови модулярних коригуючих кодів

Розроблені в розділі 2 модулярні коригуючі коди для обчислення перевірочних символів використовують взаємно прості коефіцієнти. Завдання пошуку вказаних коефіцієнтів ускладнюється тим, що необхідно знайти певну їх кількість, яка залежить від кількості інформаційних символів, при мінімальному значенні перевірного модуля.

Розроблений алгоритм для пошуку взаємно простих коефіцієнтів працює наступним чином (рисунок 3.1).

Вводимо дані (блок 1): n – кількість інформаційних символів, m – розрядність символів b_i (від 3 до 8 біт). Кількість взаємно простих коефіцієнтів V_i дорівнює кількості інформаційних символів n .

У блоці 2 вводимо модуль p взаємно простий з V_i , але не менший ніж $2 \cdot n \cdot (2^m - 1)$. У блоці 3 знаходимо взаємно прості коефіцієнти V_i .

У блоці 4 обчислюємо A_i по знайдених значеннях V_i за формулою:

$$A_i = |V_i * b_i|_p,$$

де V_i – значення взаємно простих коефіцієнтів, а b_i – значення можливих помилок.

У блоці 5 відбувається перевірка, чи обчислені значення A_i не співпадають один з одним, якщо $A_1 \neq A_i$ – знайдені дані формуємо в таблицю, якщо $A_1 = A_i$ – повторюємо всі дії доти, поки $A_1 \neq A_i$.



Рисунок 3.1 – Блок – схема алгоритму пошуку коефіцієнтів

Розглянемо приклад знаходження взаємно простих коефіцієнтів для вхідних даних: $n = 3$, $m = 3$, $p = 101$.

Після введення вхідних даних знаходимо взаємно прості числа. Перевіряємо значення A_i на співпадіння однакових чисел у кожній колонці значень. Якщо $A_1 \neq A_i$, тоді формуємо таблицю знайдених взаємно простих коефіцієнтів.

Знайдені експериментально значення взаємно простих коефіцієнтів V_i за допомогою розробленого програмного забезпечення наведені в таблиці 3.1.

Таблиця 3.1 – Значення взаємно простих коефіцієнтів V_i

V_1	V_2	V_3
2	17	79
2	41	83
3	11	13
3	37	67
3	37	71
5	17	23
5	41	47
5	47	89
5	61	73
...
61	73	79
61	79	83
61	79	97
79	83	97

Як видно з таблиці 3.2, значення A_i є різними для трьох інформаційних символів, що є необхідною умовою виправлення помилок в одному інформаційному символі.

У додатку А наведена програмна реалізація алгоритму пошуку взаємно простих коефіцієнтів для побудови модулярних коригуючих кодів.

Розроблене програмне забезпечення здійснює пошук взаємно простих коефіцієнтів для заданої кількості інформаційних символів при мінімальному значенні перевірного модуля.

Таблиця 3.2 – Результати обчислення значень A_i

V_1	e_1	A_1	V_2	e_2	A_2	V_3	e_3	A_3
2	0	0	17	0	0	79	0	0
	1	2		1	17		1	79
	2	4		2	34		2	57
	3	6		3	51		3	35
	4	8		4	68		4	13
	5	10		5	85		5	92
	6	12		6	1		6	70
	7	14		7	18		7	48
	-1	99		-1	84		-1	22
	-2	97		-2	67		-2	44
	-3	95		-3	50		-3	66
	-4	93		-4	33		-4	88
	-5	91		-5	16		-5	9
	-6	89		-6	100		-6	31
	-7	87		-7	83		-7	53

3.2 Алгоритм виявлення та виправлення помилок на основі модулярних коригуючих кодів

На основі запропонованого в пункті 2.3 методу виявлення та виправлення помилок в двох інформаційних символах розроблено алгоритм виявлення та виправлення помилок, що наведено на рисунку 3.2. У блоці 1 вводимо прийняті інформаційні символи та взаємно прості коефіцієнти. Обчислення перевірного символу по прийнятих даних відбувається в блоці 2. Значення синдрому δ (блок 3) обчислюємо як різницю між обчисленим і прийнятим перевірочними

символами. Якщо значення синдрому $\delta = 0$ (блок 4), то помилки не виявлено, а якщо $\delta \neq 0$ – помилку виявлено і переходимо до її виправлення.

Для виправлення помилки в одному символі обчислюємо значення (блок 5)

$$f_j = |v_i \cdot e_j|_p,$$

де v_i – взаємно прості коефіцієнти, e_j – значення помилки, $-2^m < e_j < 2^m$.

Для виявлення та виправлення помилки порівнюємо значення синдрому δ і розраховане значення f_j (блок 6), якщо:

– $\delta = f_j$ – помилка виявлена то її виправлення здійснюється за формулою $x_i = |x'_i \pm e_j|$ (блок 7);

– $\delta \neq f_j$ – у повідомленні є помилка більше, як в одному символі.

Припускаємо, що помилки наявні в двох символах x_i, x_j . Перевіряємо всі можливі комбінації, їх кількість визначається за формулою

$$C_n^r = \frac{n!}{r!(n-r)!} \quad (\text{блок 8}),$$

де r – кількість помилок, а n – кількість інформаційних символів.

Після цього, за розширеним алгоритмом Евкліда, знаходимо розв'язок діофантового рівняння (блок 10) та знаходимо систему розв'язків рівняння (2.13) (блок 11). Процедуру повторюємо доти, поки виконується умова (блок 9).

Перевіряємо, чи є серед знайдених розв'язків такі, що відповідають умові (блок 12).

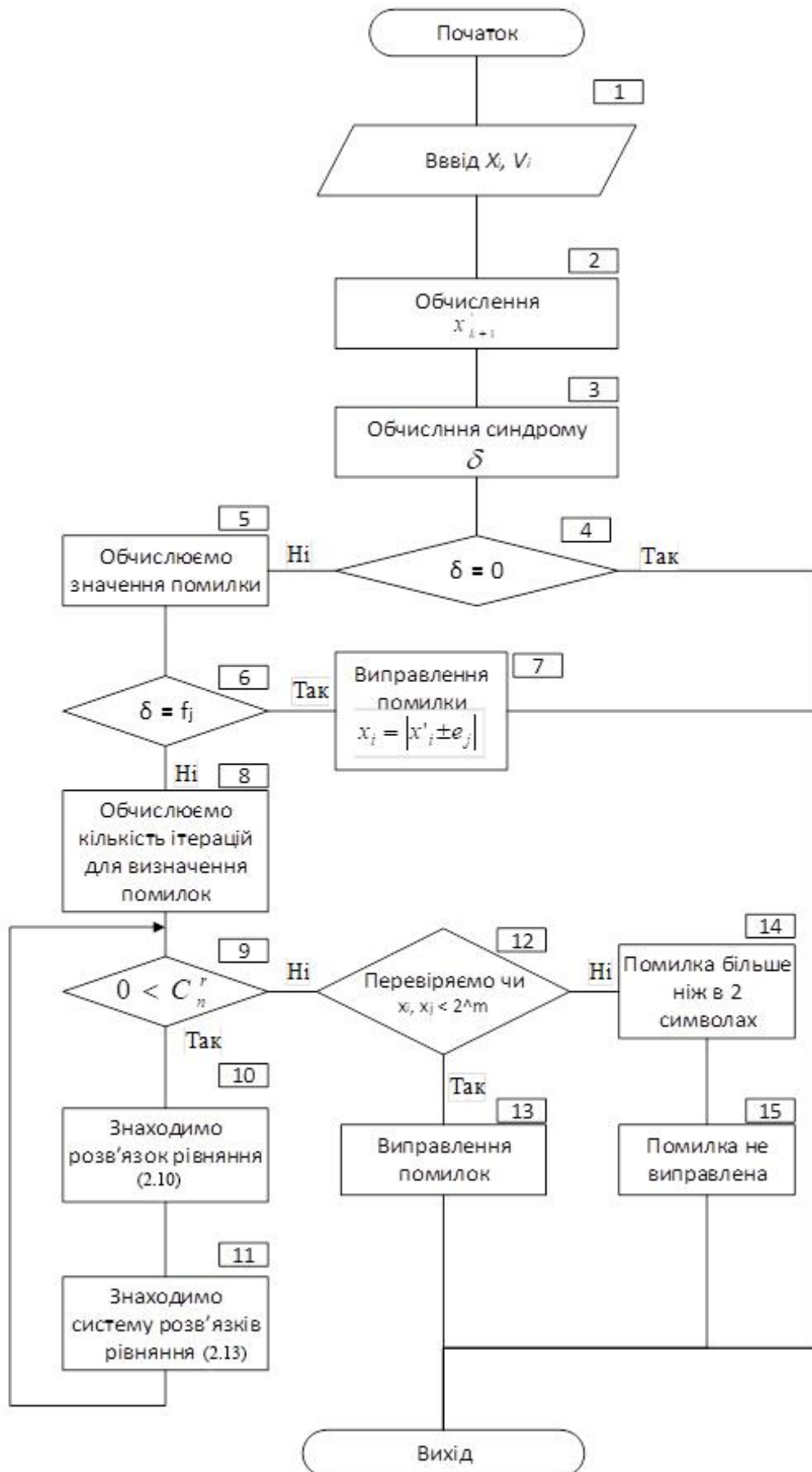


Рисунок 3.2 – Блок – схема алгоритму виявлення та виправлення помилок

$$x_i, x_j < 2^m,$$

де m – розрядність інформаційного символу.

Якщо $x_i, x_j < 2^m$, то переходимо до виправлення помилок (блок 13), якщо серед знайдених розв'язків немає правильних, то помилка є більше, як у двох інформаційних символах (блок. 14). У такому випадку помилки не будуть виправлені (блок 15), і необхідно відправляти запит на повторну передачу пакету.

Розроблена структура декодера наведена на рисунку 3.3 [72]. Визначення кількості помилок реалізовано в «блоці порівняння», у якому відбувається порівняння значення синдрому з попередньо обчисленими коефіцієнтами f_i .

Якщо $f_i = \delta$ – помилка наявна в одному символі, в іншому випадку – помилка в двох і більше символах (див. рисунок 3.3).

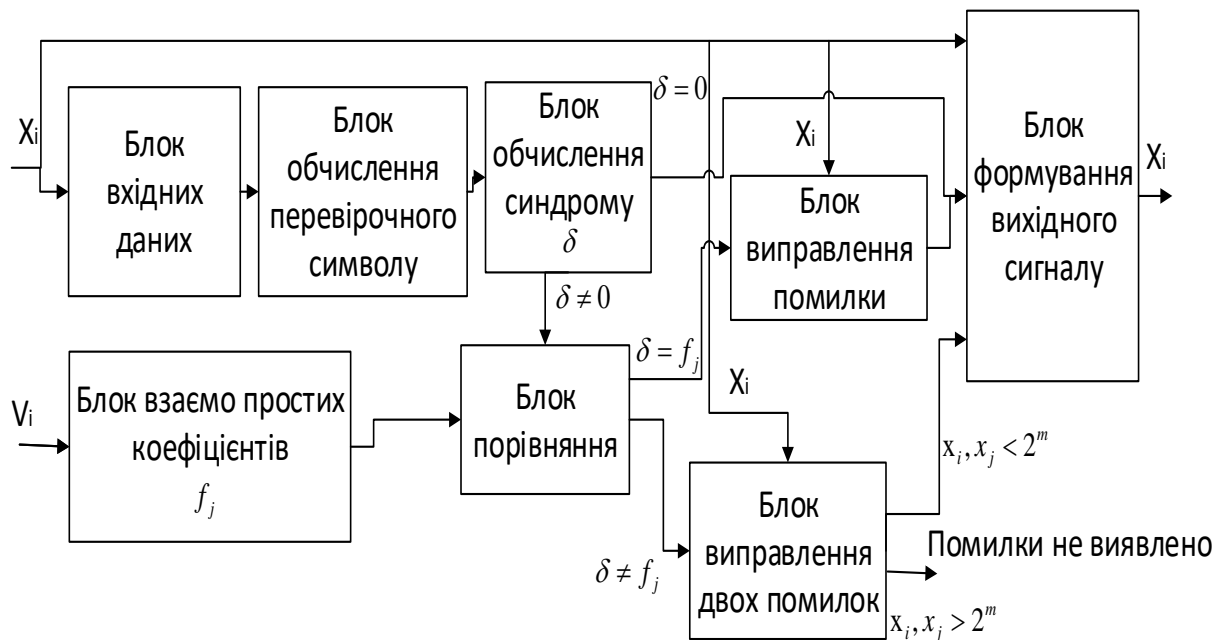


Рисунок 3.3 – Структурна схема декодера

Розроблений метод виправлення помилок планується застосовувати для підвищення ефективності безпроводних сенсорних мереж на основі інтеграції каналного кодування і завадостійкого кодування даних за допомогою модулярних коригуючих кодів та двовимірної контролю помилок.

3.3 Алгоритм виявлення та виправлення пакетів помилок на основі модулярних коригуючих кодів

На основі запропонованого у другому розділі методу виправлення пакетів помилок розроблено алгоритм виявлення та виправлення пакетів помилок з використанням модулярних коригуючих кодів та двовимірної схеми.

Алгоритм працює наступним чином (рисунок 3.4): у блоці 1 вводимо взаємно прості коефіцієнти v_i , значення модуля P та інформаційні символи x_i . Далі в блоці 2 і блоці 3 відбувається обчислення синдромів по рядках та по стовпцях за формулами (див. 2.15, 2.16).

Якщо значення синдрому (блок 4) $\delta_i = 0, \delta_j = 0$, то помилки не виявлено, а якщо $\delta_i \neq 0, \delta_j \neq 0$ або якщо хоч в одному із синдромів $\delta_i = 0, \delta_j \neq 0$, то помилку виявлено і переходимо до її виправлення.

Для виправлення помилки (блок 5) в одному символі для кожного рядка розв'язуємо рівняння

$$v_i(x'_i - x_i) = \delta_i \pmod{P},$$

де v_i – взаємно прості коефіцієнти, x'_i – прийняте повідомлення та x_i – повідомлення, яке відправлялося.

Після того, як знайшли розв'язки даного рівняння для всіх рядків, перевіряємо значення синдрому (блок 6) при: $\delta_i \neq 0, \delta_j = 0$ – помилки є в стовпцях, які виправляємо за формулою (див. 2.18) (блок 8), та в блоці 9

підставляємо значення розв'язків у матрицю; при $\delta_i = 0, \delta_j \neq 0$ – помилки є в рядках, які виправляємо за формулою (див. 2.18) (блок 7).

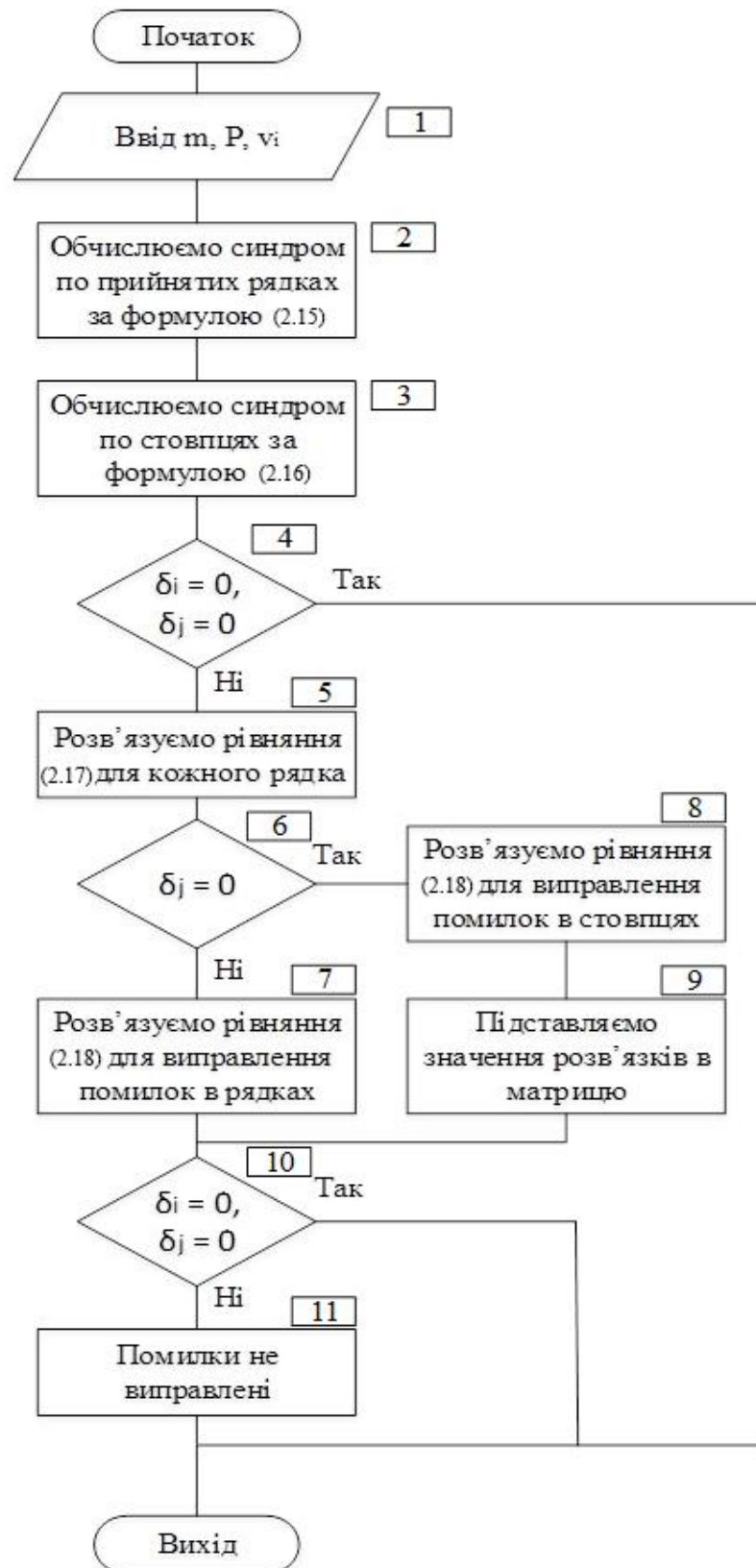


Рисунок 3.4 – Блок – схема алгоритму виявлення та виправлення пакетів
ПОМИЛОК

У блоці 10 робимо перевірку, і якщо значення синдрому $\delta_i = 0, \delta_j = 0$, то помилки виправлені, а якщо значення синдрому $\delta_i \neq 0, \delta_j \neq 0$ – помилок більше, ніж алгоритм може виправити (блок 11).

Розглянемо приклад. Побудуємо двовимірний код на основі модулярного коригуючого коду при $k = 8, l = 8$, розрядність інформаційних символів $m = 4$, модуль $P = 1021$, коефіцієнти: $v_1 = 13, v_2 = 17, v_3 = 19, v_4 = 23, v_{15} = 29, v_6 = 31, v_7 = 37, v_8 = 43$ (таблиця 3.3).

Таблиця 3.3 – Модулярний двовимірний код на виході передавача

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Перевір ка по рядках
	3	4	9	10	12	14	6	8	835
	11	14	15	0	1	2	7	4	167
	4	7	2	15	12	3	11	13	940
	1	2	3	4	7	8	10	0	1017
	2	4	6	7	1	8	11	15	677
	5	4	1	3	7	9	7	3	70
	3	8	10	14	12	5	6	7	692
	12	11	7	4	3	2	1	8	77
Перевірка по стовпцях	144	457	334	472	360	218	416	529	

Розглянемо алгоритм виправлення помилок при максимальній кількості спотворених символів (таблиця 3.4).

Таблиця 3.4 – Модулярний двовимірний код на вході приймача

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Перевір ка по рядках
	12*	11*	6*	5*	3*	1*	9*	7*	303
	4*	1*	0*	15*	14*	13*	8*	11*	971
	4	7	13*	15	12	3	11	13	128
	1	2	12*	4	7	8	10	0	167
	2	4	9*	7	1	8	11	15	734
	5	4	14*	3	7	9	7	3	317
	3	8	5*	14	12	5	6	7	597
	12	11	8*	4	3	2	1	8	96
Перевірка по стовпцях	142	327	804	662	464	236	472	635	

* – спотворені символи.

Значення синдромів обчислені по рядках і стовпцях матриці даних наведені в таблиці 3.5.

Таблиця 3.5 – Значення синдрому по рядках і стовпцях

Синдром по рядках, δ_i	Синдром по стовпцях, δ_j
489	1019
804	891
209	470
171	190
57	104
247	18
926	56
19	106

Оскільки, синдроми не дорівнюють нулю, то спотворені символи ϵ в усіх рядках та стовпцях матриці даних (див. таблицю 3.5).

Алгоритм виявлення та виправлення помилок складається з наступних кроків.

1. Перевіряємо чи ϵ рядки з однією помилкою. Для цього послідовно розв'язуємо рівняння (2.17) для кожного рядка. Результати розв'язку наведені в таблиці 3.6.

Таблиця 3.6 – Результати розв'язку рівняння 2.15

№ рядка	Інформаційні символи							
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	52,92	42,29	34	28,13	21,34	18,16	23,37	19,37
2	20,69	13,76	11,42	24,43	21,48	20	13,86	16,04
3	66,46	54,76	2	5,91	4,79	29,19	5,35	8,13
4	66,38	52	3	40,96	1,1	2,48	5,37	19,76
5	76,15	0,65	6	4,52	34,24	6,16	9,46	13,67
6	64,54	49,53	1	36,65	33,69	1,04	0,32	21
7	10,31	13,59	10	18,13	15,28	8,06	8,56	9,21
8	10,54	9,88	7	3,17	2,34	1,39	0,48	7,56

Як видно з таблиці 3.6, знайдені розв'язки, цілі числа в заданому діапазоні [0–15] свідчать про те, що спотворений символ x_3 в 3 – 8 рядках.

2. Підставивши обчислені значення в таблицю 3.4, отримаємо нову таблицю синдромів (таблиця 3.7).

Аналіз таблиці 3.7 показує, що помилки ϵ в усіх стовпцях але тільки у двох рядках. Тобто, спотворені всі символи в першому і другому рядку. Виправлення помилок в двох символах здійснюється за допомогою рівняння (2.18).

Таблиця 3.7 – Таблиця синдромів після часткового виправлення помилок

Синдром по рядках, δ_i	Синдром по стовпцях, δ_j
489	1019
804	891
0	727
0	190
0	104
0	18
0	56
0	106

Для виправлення помилок у символах, розміщених в першому стовпці, підставимо числові значення в рівняння (2.18) і отримаємо:

$$\begin{aligned} 13(12 - x_1) + 17(4 - x_2) &= 1019(\text{mod } 1021); \\ 13x_1 + 17x_2 &= 226(\text{mod } 1021); \end{aligned} \quad (3.1)$$

Розв'язавши рівняння (3.1) для кожного стовпця матриці, знайдемо правильні значення інформаційних символів (таблиця 3.8).

Як видно з таблиці 3.8, знайдені розв'язки рівняння (3.1) однозначно забезпечують виправлення помилок в 22 символах вхідного повідомлення. Отже, розроблений код дозволяє виправляти пакети помилок з максимальною довжиною $b = 3k - 2$ символів, при розміщенні спотворених символів в двох рядках і одному стовпці, або в двох стовпцях і одному рядку. При цьому забезпечує швидкість коду $R_1 = \frac{k_1}{n_1} = 0.62$, при $k = 8$, $m_1 = 4$, $P = 1021$, $k_1 = k * k * m_1 = 256$, $r_1 = 2 * k * \lceil \log_2 P \rceil = 160$, $n_1 = k_1 + r_1 = 416$ та $R_2 = \frac{k_2}{n_2} = 0.67$ при $m_2 = 8$, $k_2 = 512$, $n_2 = 768$. Відносна надлишковість коду дорівнює: $r = (1 - R_1) * 100\% = 38\%$.

Таблиця 3.8 – Розв’язки рівняння 3.1

	Інформаційні символи							
x_2	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}
0	17,38	22,31	28,62	10	13,31	16,62	15,15	13,23
1	16,08	21	27,31	8,69	12	15,31	13,85	11,92
2	14,77	19,69	26	7,38	10,69	14	12,54	10,62
3	13,46	18,38	24,69	6,08	9,38	12,69	11,23	9,31
4	12,15	17,08	23,38	4,77	8,08	11,38	9,92	8
5	10,85	15,78	22,08	3,46	6,77	10,08	8,62	6,69
6	9,54	14,46	20,77	2,15	5,46	8,77	7,31	5,38
7	8,23	13,15	19,46	0,85	4,15	7,46	6	4,08
8	6,92	11,84	18,15	-0,46	2,85	6,15	4,69	2,77
9	5,62	10,54	16,85	-1,77	1,54	4,85	3,38	1,46
10	4,31	9,23	15,54	-3,08	0,23	3,54	2,08	0,15
11	3	7,92	14,23	-4,38	-1,08	2,23	0,77	-1,15
12	1,69	6,62	12,92	-5,69	-2,38	0,92	-0,54	-2,46
13	0,38	5,31	11,62	-7,00	-3,69	-0,38	-1,85	-3,77
14	-0,92	4	10,31	-8,31	-5,00	-1,69	-3,15	-5,08
15	-2,23	2,69	9	-9,62	-6,31	-3,00	-4,46	-6,38

Розроблено алгоритм виявлення та виправлення випадкових однократних помилок та пакетів помилок максимальною довжиною $b = 3k - 2$ при швидкості коду $R = 0.67$.

Враховуючи низьку складність реалізації алгоритму декодування, даний код планується використати для підвищення надійності передачі даних у безпроводних сенсорних мережах.

3.4 Алгоритми виправлення помилок на основі коригуючих кодів системи залишкових класів

Розроблено алгоритми на основі запропонованого в пункті 2.1 методу виправлення помилок з використанням коригуючих кодів системи залишкових класів зі спеціальною системою модулів.

3.4.1 Алгоритм послідовного обчислення проекції числа

Розроблений алгоритм для послідовного обчислення проекції числа представлений на рисунку 3.5.

Алгоритм працює наступним чином. У блоці 1 здійснюється ввід даних (блок 1), де: x_i – прийняте повідомлення, яке i є залишком по вибраних модулях p_i ; p_i – спеціальні модулі, які вибираються для різної розрядності вхідних даних; P_k – робочий діапазон модулів; B_i – базисні числа для вибраного набору модулів.

У блоці 2 відбувається обчислення позиційного представлення числа X . У блоці 3 здійснюється порівняння позиційного числа X з робочим діапазоном P_k . Якщо значення не виходить за межі робочого діапазону ($X < P_k$), то помилки немає, і переходимо до блоку 4, а якщо значення виходить за межі робочого діапазону ($X > P_k$), то це означає, що помилка виявлена.

Для подальшого виправлення помилок у блоці 5 послідовно обчислюються проекції чисел з відкиданням одного модуля. У блоці 6 – відбувається виправлення помилки.

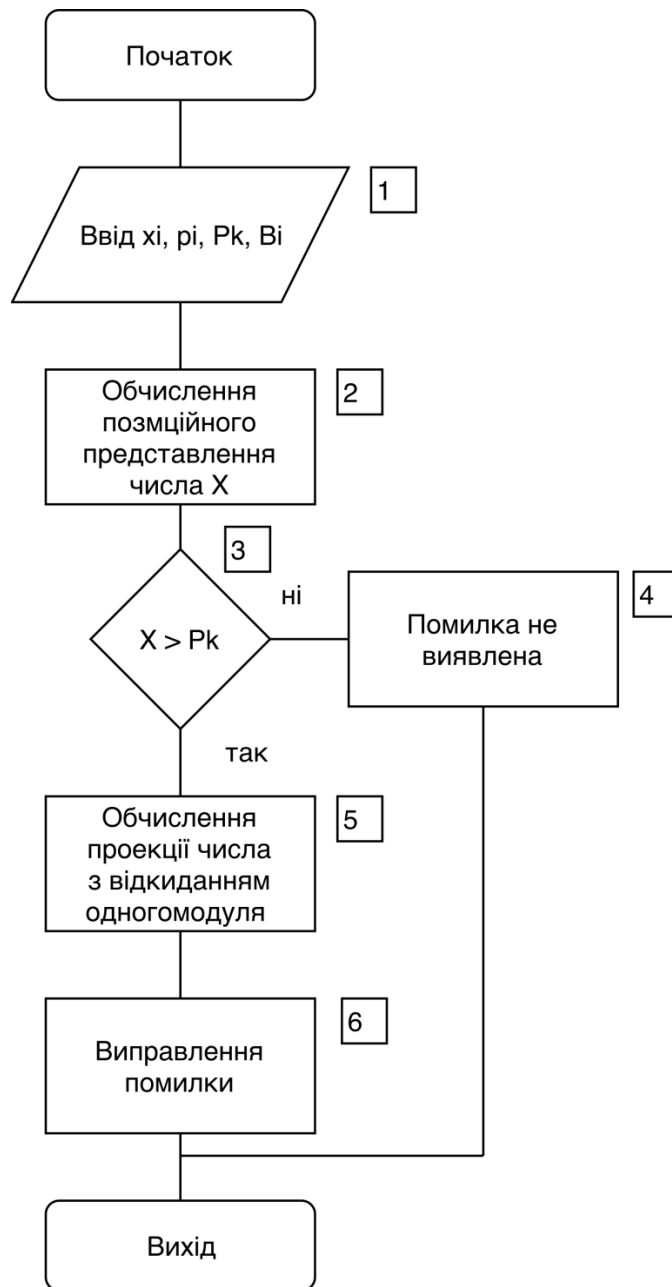


Рисунок 3.5 – Блок – схема алгоритму виявлення та виправлення помилок на основі обчислення проєкції числа

Даний алгоритм здійснює виявлення та виправлення помилок за допомогою аналізу обчислених проєкцій числа.

3.4.2 Алгоритм виявлення та виправлення помилок з використанням таблиць синдрому

Розроблений алгоритм для виявлення та виправлення помилок з використанням таблиць синдрому представлений на рисунку 3.6. У блоці 1 вводиться вхідне повідомлення, два перевірючі символи і модулі.

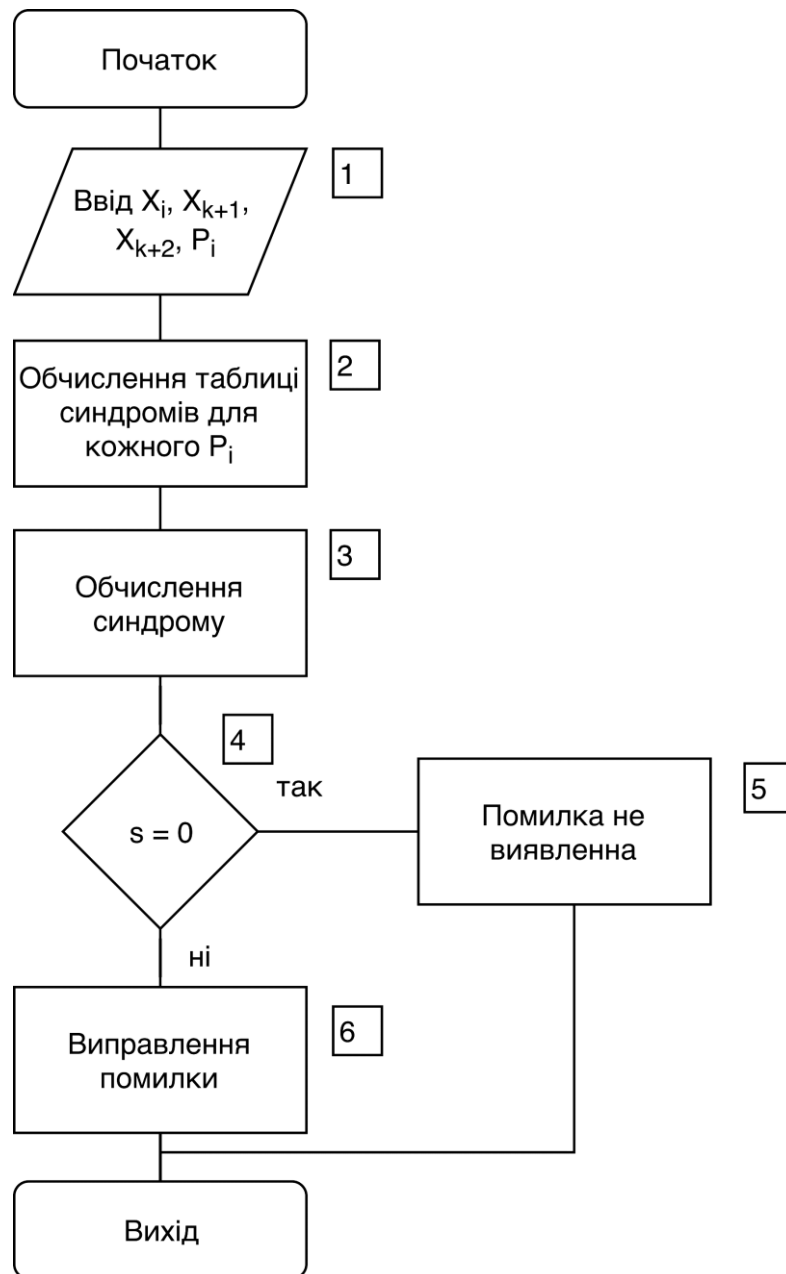


Рисунок 3.6 – Блок – схема алгоритму виявлення помилки з використанням таблиці синдрому

Попередньо у блоці 2 обчислюється (або зберігається в пам'яті) синдром для кожного з модулів p_i та будується таблиця синдрому, по якій буде здійснюватися виправлення помилок (таблиця 3.9).

Таблиця 3.9 – Значення синдрому та помилки за модулями

№	p_1		№	p_2		...	№	p_5		№	p_6	
	s	e_i		s	e_i			s	e_i		s	e_i
0	2172	12	0	6483	2	...	0	1270	30	0	7085	27
1	2353	13	1	7045	3	...	1	4064	31	1	4290	28
2	2534	14	2	4711	4	...	2	6858	32	2	1495	29
3	2715	15	3	5273	5	...	3	1397	33	3	6955	30
5	181	1	4	2939	6	...	4	4191	34	4	4160	31
6	362	2	5	3501	7	...	5	6985	35	5	1365	32
7	543	3	6	1167	8	...	6	1524	36	6	6825	33
8	724	4	7	1729	9	...	7	4318	37	7	4030	34
9	905	5	8	7650	10	...	8	7112	38	8	1235	35
10	1086	6	9	8212	11	...	9	1651	39	9	6695	36
11	1267	7	10	5878	12	...	10	4445	40	10	3900	37
12	1448	8	11	6440	13	...	11	7239	41	11	1105	38
13	1629	9	12	4106	14	...	12	1778	42	12	6565	39
14	1810	10	13	4668	15	...	13	4572	43	13	3770	40
15	1991	11	14	2334	16	...	14	7366	44	14	975	41
			16	562	18
						...	63	3937	93	125	4420	152
						...	64	6731	94	126	1625	153

Наступним кроком (блок 3) обчислюється синдром s для виявлення помилки за формулою (2.2).

У блоці 4 проводиться перевірка синдрому, якщо синдром дорівнює нулю ($s = 0$) – помилка відсутня, якщо синдром не дорівнює нулю ($s \neq 0$) – помилка виявлена.

При $s \neq 0$ у блоці 6 відбувається виправлення помилки. Оскільки різним помилкам відповідає різне значення синдрому, то для виявлення помилок використовуємо попередньо сформовані таблиці синдромів для кожного модуля p_i .

Для виправлення помилки використовуємо таблицю синдромів, що відповідає певному значенню помилки, та знаходимо правильне значення інформаційного символу.

На основі розроблених алгоритмів у розділі 4.1 реалізовано пристрої виявлення та виправлення помилок на основі послідовного обчислення проекції числа та з використанням таблиць синдрому.

Висновки до третього розділу

1. Розроблено алгоритм пошуку взаємно простих коефіцієнтів для обчислення перевірочних символів при побудові модулярних коригуючих кодів.

2. Розроблено метод виправлення помилок у двох інформаційних символах при використанні одного перевірочного символу на основі модулярних коригуючих кодів. Розроблено структурну схему декодера виявлення та виправлення помилок.

3. Розроблено алгоритм виявлення та виправлення пактів помилок на основі модулярних коригуючих кодів та двовимірної схеми. Даний алгоритм забезпечує виправлення пакетів помилок максимальної довжини $b = 3k - 2$ при розміщенні помилок у двох рядках і одному стовпці або в двох стовпцях і одному рядку.

4. Розроблені алгоритми для виявлення та виправлення помилок за допомогою послідовного обчислення проекції числа та з використанням таблиць синдрому.

РОЗДІЛ 4

СИНТЕЗ ПРИСТРОЇВ ЗАВАДОСТІЙКОГО КОДУВАННЯ ТА ПЕРЕДАВАННЯ ДАНИХ В БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖАХ НА ПРОГРАМОВАНИХ ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМАХ

4.1 Пристрій виправлення помилок на основі коригуючих кодів системи залишкових класів зі спеціальною системою модулів

На основі запропонованого в пункті 2.1 методу вибору розширеної системи спеціальних модулів при формуванні коригуючих кодів системи залишкових класів розроблено загальні структури та реалізовано на ПЛІС декодери на основі двох алгоритмів виправлення помилок:

- 1) послідовне обчислення проекції числа за модулем;
- 2) виправлення помилки з використанням попередньо обчислених таблиць синдрому.

Спеціальна система модулів сформована із чисел, обчислених за формулами: 2^n , $2^n + 1$, $2^{n+1} - 1$, $2^{n+2} - 1$, $2^{n+2} + 1$, $2^{n+3} - 1$ при різному значенні n .

У таблиці 4.1 наведені спеціальні модулі для вхідної розрядності даних 16, 24 і 32 розряди.

Основними вимогами при виборі модулів для синтезу декодера є:

- взаємна простота модулів;
- добуток інформаційних модулів повинен бути більшим за 2^{24} ;
- максимальна швидкість коду.

Таблиця 4.1 – Набір спеціальних модулів для різної розрядності вхідних даних

Розрядність повідомлення	Інформаційні модулі							Перевірочні модулі		Швидкість коду, $R = k/n$	
	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	R	
16	1	17	31	127	-	-	-	-	128	129	0,53
	2	5	7	31	64	-	-	-	127	129	0,55
24	1	17	32	127	257	-	-	-	511	513	0,58
	2	5	7	31	127	128	-	-	129	513	0,6
	3	2	17	31	127	129	-	-	513	1025	0,56
	4	5	17	31	64	127	-	-	129	257	0,61
	5	3	5	7	31	64	127	-	129	257	0,61
32	1	17	31	127	256	257	-	-	511	513	0,65
	2	3	5	7	31	64	127	257	2047	4097	0,59

Для реалізації декодера вибрано систему модулів: 17, 31, 127, 256, 257, 511, 513, з яких 17, 31, 127, 256, 257 – інформаційні; 511, 513 – перевірочні. Даний набір модулів забезпечує виправлення помилок у залишках з будь-якого одного модуля та максимальну швидкість коду при розрядності вхідних даних 32 біти (див. таблицю 4.1). Вибір системи модулів з максимальною швидкістю коду забезпечить мінімальну надлишковість коригуючих кодів.

4.1.1 Декодер виявлення та виправлення помилок на основі методу обчислення проекції числа

Розроблений декодер виявлення та виправлення помилок на основі методу обчислення проекції числа складається з блоку виявлення помилок, блоку обчислення проекцій числа та блоку виправлення помилок (рисунок 4.1).

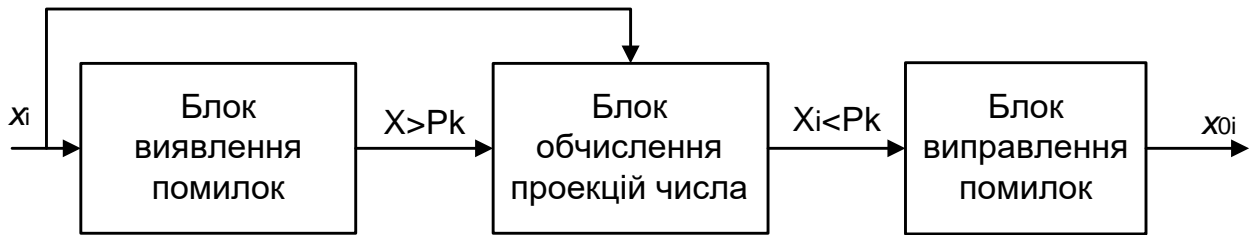


Рисунок 4.1 – Загальна структурна схема декодера виявлення та виправлення помилок на основі обчислення проєкції числа

У блоці виявлення помилок відбувається обчислення позиційного значення отриманого повідомлення X та порівняння його з робочим діапазоном P_k . Якщо отримане значення виходить за межі робочого діапазону ($X > P_k$) – помилка виявлена і сигнал подається в блок обчислення проєкцій числа для локалізації помилки та виправлення (див. рисунок 4.1):

$$X = \left| \sum_{i=0}^n (x_i \cdot B_i) \right|_{P_g}, \quad (4.1)$$

де B_i – базисні числа, x_i – залишки за вибраними модулями, P_g – загальний діапазон, n – кількість модулів.

Загальний діапазон обчислюється за формулою:

$$P_g = \prod_{i=0}^n p_i,$$

для вибраних модулів дорівнює $P_g = 1154318590381824$.

Робочий діапазон:

$$P_k = \prod_{i=0}^k p_i,$$

$$P_k = 4403392768.$$

Базисні числа для вибраного набору модулів дорівнюють:

$$B_0 = 135802187103744;$$

$$B_1 = 930902089017600;$$

$$B_2 = 254495437249536;$$

$$B_3 = 500505326298369;$$

$$B_4 = 49406632273152;$$

$$B_5 = 833549040804096;$$

$$B_6 = 758295058398976.$$

Для розпаралелювання процесу обчислення позиційного представлення числа X формулу (4.1) запишемо у вигляді:

$$X = |f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6|_{P_g},$$

де

$$f_0 = |(x_0 \cdot B_0)|_{P_g};$$

$$f_1 = |(x_1 \cdot B_1)|_{P_g};$$

...

$$f_6 = |(x_6 \cdot B_6)|_{P_g}.$$

Виправлення помилок відбувається в два етапи. На першому етапі послідовно обчислюються проекції чисел з відкиданням одного модуля (див.рисунок 4.1). На другому етапі відбувається виправлення помилок.

Проекції числа за кожним модулем обчислюються за формулами:

$$X_0 = |(x_{01} \cdot B_{01} + x_{02} \cdot B_{02} + x_{03} \cdot B_{03} + x_{04} \cdot B_{04} + x_{05} \cdot B_{05} + x_{06} \cdot B_{06})|_{P_0};$$

$$X_1 = |(x_{00} \cdot B_{10} + x_{02} \cdot B_{12} + x_{03} \cdot B_{13} + x_{04} \cdot B_{14} + x_{05} \cdot B_{15} + x_{06} \cdot B_{16})|_{P_1};$$

$$X_2 = |(x_{00} \cdot B_{20} + x_{01} \cdot B_{21} + x_{03} \cdot B_{23} + x_{04} \cdot B_{24} + x_{05} \cdot B_{25} + x_{06} \cdot B_{26})|_{P_2};$$

$$X_3 = |(x_{00} \cdot B_{30} + x_{01} \cdot B_{31} + x_{02} \cdot B_{32} + x_{04} \cdot B_{34} + x_{05} \cdot B_{35} + x_{06} \cdot B_{36})|_{P_3};$$

$$X_4 = |(x_{00} \cdot B_{40} + x_{01} \cdot B_{41} + x_{02} \cdot B_{42} + x_{03} \cdot B_{43} + x_{05} \cdot B_{45} + x_{06} \cdot B_{46})|_{P_4};$$

$$X_5 = |(x_{00} \cdot B_{50} + x_{01} \cdot B_{51} + x_{02} \cdot B_{52} + x_{03} \cdot B_{53} + x_{04} \cdot B_{54} + x_{06} \cdot B_{56})|_{P_5};$$

$$X_6 = |(x_{00} \cdot B_{60} + x_{01} \cdot B_{61} + x_{02} \cdot B_{62} + x_{03} \cdot B_{63} + x_{04} \cdot B_{64} + x_{05} \cdot B_{65})|_{P_6};$$

де B_{ij} – базисні числа для проекції; i – номер проекції; j – номер базисного числа в проекції; P_i – добуток модулів без включення i – го модуля;

$$P_0 = p_1 \cdot p_2 \cdot p_3 \cdot p_4 \cdot p_5 \cdot p_6;$$

$$P_1 = p_0 \cdot p_2 \cdot p_3 \cdot p_4 \cdot p_5 \cdot p_6;$$

$$P_2 = p_0 \cdot p_1 \cdot p_3 \cdot p_4 \cdot p_5 \cdot p_6;$$

$$P_3 = p_0 \cdot p_1 \cdot p_2 \cdot p_4 \cdot p_5 \cdot p_6;$$

$$P_4 = p_0 \cdot p_1 \cdot p_2 \cdot p_3 \cdot p_5 \cdot p_6;$$

$$P_5 = p_0 \cdot p_1 \cdot p_2 \cdot p_3 \cdot p_4 \cdot p_6;$$

$$P_6 = p_0 \cdot p_1 \cdot p_2 \cdot p_3 \cdot p_4 \cdot p_5.$$

Виправлення помилок здійснюється на основі аналізу обчислених проекцій, при цьому, якщо проекція $X_i < P_k$ за модулем p_i , то спотворений залишок за даним модулем. Відповідно, виправлення помилки полягає в отриманні залишку від проекції за даним модулем $x_{0i} = |X_i|_{p_i}$.

Алгоритм виявлення та виправлення помилок з використанням спеціальної системи модулів написано на мові Verilog. Функціональна схема декодера наведена на рисунку 4.2.

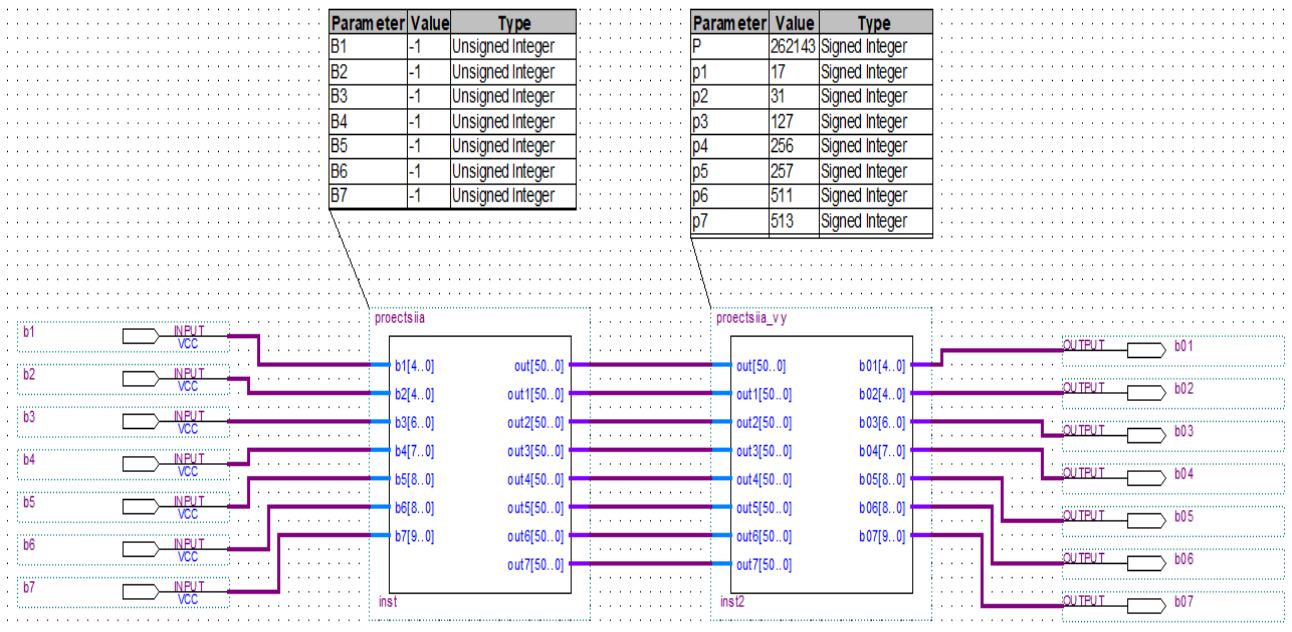


Рисунок 4.2 – Функціональна схема декодера

Моделювання та синтез пристрою проведено в середовищі Quartus II фірми Altera. Пристрій реалізовано на ПЛІС серії Cyclone IV GX, мікросхема EP4CGX150CF23C7.

Результати моделювання роботи декодера показали його коректну роботу (рисунок 4.3).

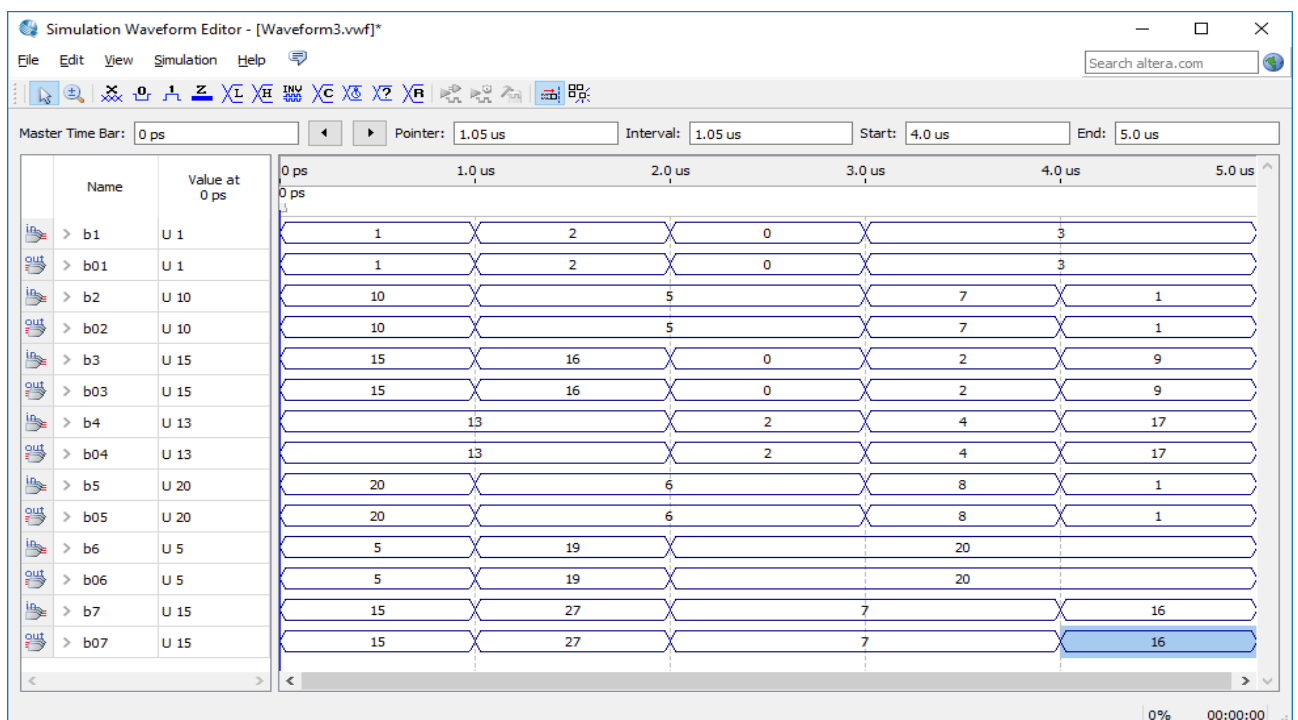


Рисунок 4.3 – Результати симуляції роботи декодера: b1-b7– входи; b01-b07 – виходи

4.1.2 Декодер виправлення помилок на основі алгоритму обчислення синдрому

Декодер виправлення помилок на основі обчислення синдрому базується на обчисленні різниці між позиційним представленням числа, обчисленим за інформаційними залишками, і позиційним представленням, обчисленим за перевірочними залишками [113, 60, 107]:

$$\delta = \left| |X'|_{P_k} - |X'|_{P_r} \right|_{P_r},$$

де $|X'|_{P_k} \equiv (x'_1, x'_2, \dots, x'_k)$, $|X'|_{P_r} \equiv (x'_{k+1}, x'_{k+2})$, P_k – робочий діапазон, P_r – перевірочний діапазон.

На рисунку 4.4 представлена узагальнена структура декодера, яка складається з блоків обчислення позиційного представлення даних за інформаційними $(x'_1, x'_2, \dots, x'_k)$ та перевірочними (x'_{k+1}, x'_{k+2}) модулями, блоку обчислення синдрому, таблиці відповідності можливих помилок значенню синдрому та блоку виправлення помилки.

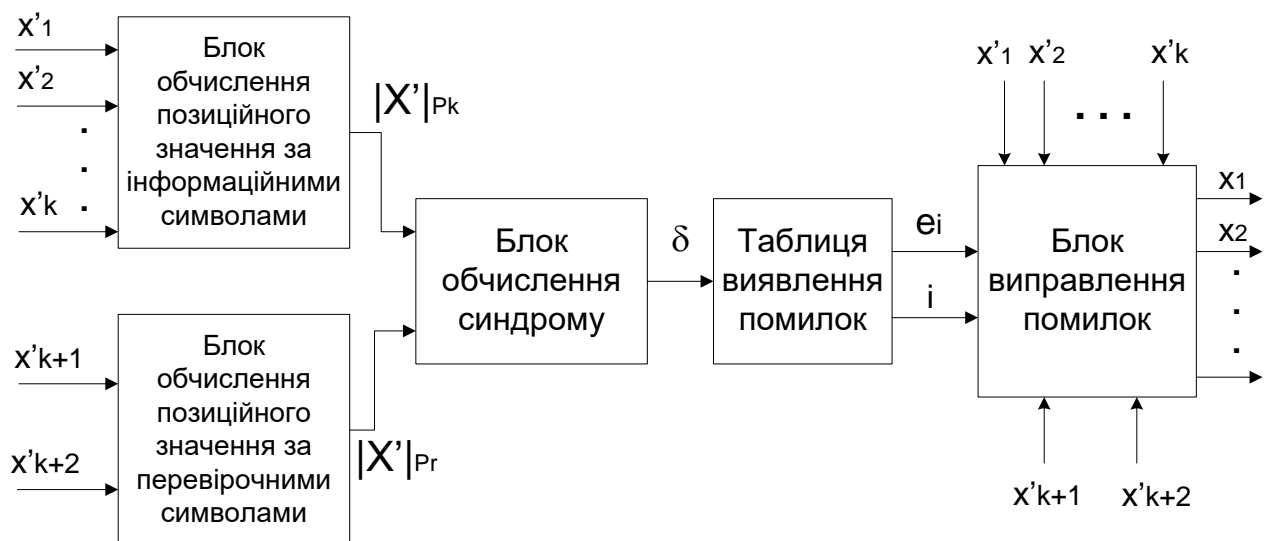


Рисунок 4.4 – Структурна схема декодера

У таблиці виявлення помилок (див. рисунок 4.4) зберігаються значення помилок, які однозначно відповідають значенню синдрому δ .

У блоці виправлення помилок (див. рисунок 4.4) на основі значень помилок і прийнятих даних відбувається виправлення помилок.

Оскільки для обчислення синдрому використовується робочий та перевірочний діапазон, то розрядність чисел буде меншою приблизно на 10 -13 розрядів порівняно з методом проєкцій, який використовує повний діапазон.

Декодер описаний на мові Verilog у середовищі Quartus II та реалізований на ПЛІС фірми «Altera» серії Arria 10, пристрій 10AX115R4F40I3SG. Результати синтезу декодера наведені на рисунку 4.5.


Flow Summary	
 <<Filter>>	
Flow Status	Successful - Tue Aug 22 14:56:42 2017
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Standard Edition
Revision Name	syndrom32
Top-level Entity Name	syndrom32
Family	Arria 10
Device	10AX115R4F40I3SG
Timing Models	Final
Logic utilization (in ALMs)	321,966 / 427,200 (75 %)
Total registers	0
Total pins	126 / 650 (19 %)
Total virtual pins	0
Total block memory bits	0 / 55,562,240 (0 %)
Total DSP Blocks	0 / 1,518 (0 %)
Total HSSI RX channels	0 / 66 (0 %)
Total HSSI TX channels	0 / 66 (0 %)
Total PLLs	0 / 138 (0 %)

Рисунок 4.5 – Параметри синтезу декодера

Результати симуляції, наведені на рисунку 4.6, показують виявлення та виправлення помилки, що підтверджує коректну роботу декодера.

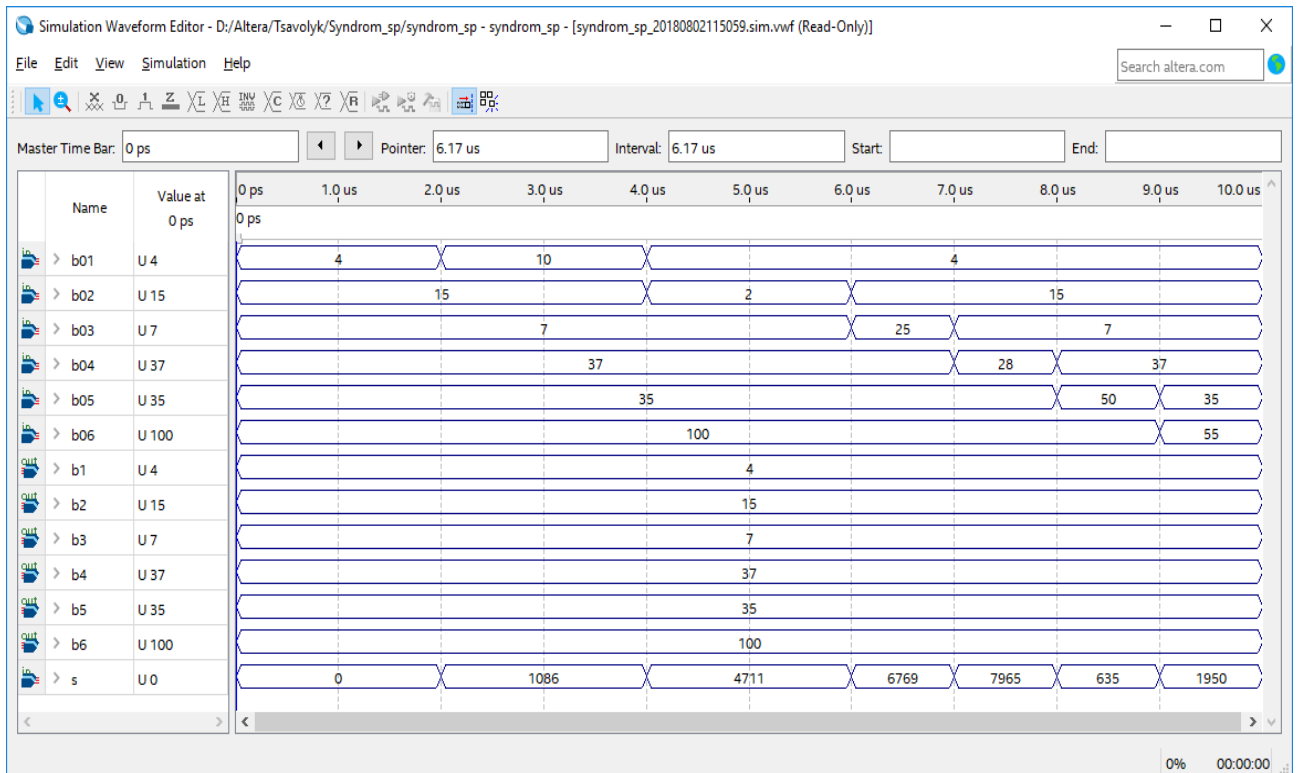


Рисунок 4.6 – Результати симуляції роботи декодера: b01-b06 – входи;
b1-b7 – виходи

Проведені експериментальні дослідження показали, що коригуючі коди системи залишкових класів зі спеціальною системою модулів забезпечують зменшення апаратної складності та підвищення швидкодії при реалізації кодерів/декодерів на програмованих логічних інтегральних схемах як на етапі отримання залишків, так і на етапі виявлення та виправлення помилок у методі обчислення проєкцій.

Однак через необхідність зберігання таблиць «синдром – значення помилки» апаратна складність реалізації на ПЛІС алгоритму виправлення помилок на основі методу обчислення проєкцій є меншою приблизно в 3,6 рази (обчислення проєкцій – 88140 лог. елем., обчислення синдрому – 321966 лог. елем.) порівняно з методом обчислення синдрому.

4.2 Пристрій обчислення перевірочних символів в коригуючих кодах системи залишкових класів

На основі запропонованого в пункті 2.2 нового методу формування коригуючих кодів СЗК, розроблено структуру та реалізовано на ПЛІС пристрій кодування (рисунок 4.7).

Процес кодування складається з трьох етапів (див. рисунок 4.7):

- 1) розділення вхідного повідомлення на k частин (4 – 8 біт);
- 2) переведення повідомлення в позиційну систему числення у вибраній системі модулів;
- 3) обчислення перевірочних символів.

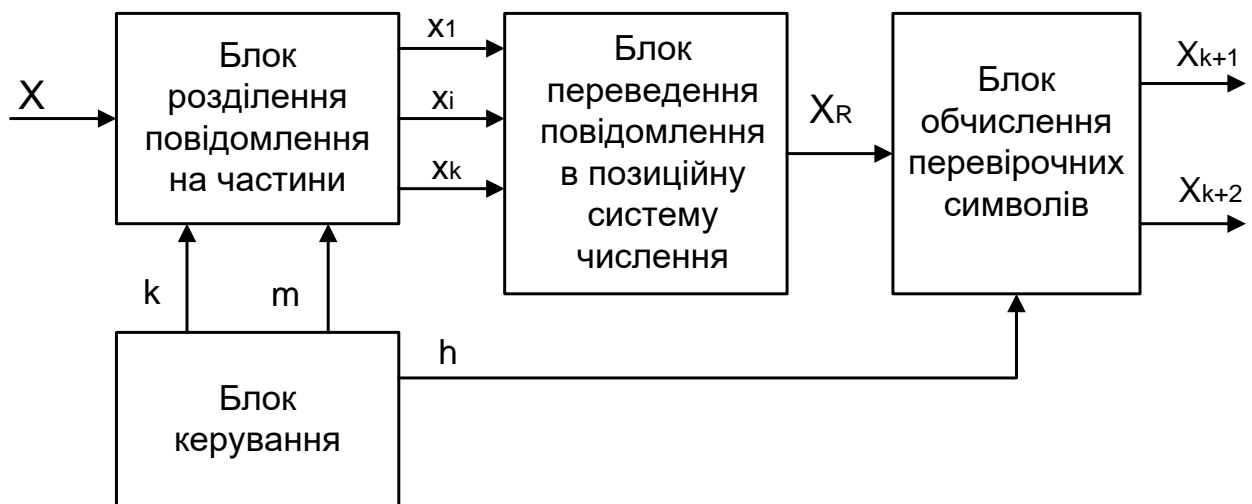


Рисунок 4.7 – Узагальнена структура пристрою кодування

Процес розділення вхідного повідомлення полягає в присвоєнні змінним $x_1, x_2, \dots, x_i, \dots, x_k$ m - біт повідомлення X , де k – кількість інформаційних модулів.

При $m = 8$ і $k = 4$ процес розділення на мові Verilog має вигляд:

```

input [31:0] X; # вхідне повідомлення
assign x1 = X [7:0];
assign x2 = X [15:8];
  
```

```
assign x3 = X [23:16];
assign x4 = X [31:24].
```

При синтезі на ПЛІС описана операція не вимагає апаратних затрат.

В якості інформаційних модулів вибрано прості числа: $p_1=257$, $p_2=263$, $p_3=269$, $p_4=271$. Перевірочні модулі: $p_5=277$, $p_6=281$.

Базисні числа рівні:

```
B1 = 3201796979,
B2 = 393435903,
B3 = 3113917370,
B4 = 3145482367.
```

Робочий діапазон $P = 4927316309$.

Процедура обчислення позиційного значення повідомлення:

```
assign A1 = (x1*B1)%P;
assign A2 = (x2*B2)%P;
assign A3 = (x3*B3)%P;
assign A4 = (x4*B4)%P;
assign XR = (A1+A2+A3+A4)%P.
```

У блоці обчислення перевірочних символів відбувається обчислення залишків за перевірочними модулями (див. рисунок 4.8):

```
assign x_5 = XR %p5;
assign x_6 = XR %p6.
```

Для розширення функціональних можливостей кодера в блоці керування задається розрядність повідомлення – m , кількість інформаційних модулів – k та кількість перевірочних модулів – h (див. рисунок 4.8). На виходах

пристрою $x_{k+1}[8:0]$, $x_{k+2}[8:0]$ формуються перевірочні символи залежно від вхідного значення X_R (h – вказує кількість перевірочних модулів 1 або 2) та перевірочних модулів. Розрядність вхідного повідомлення задається кодом на вході k відповідно до таблиці 4.2.

Таблиця 4.2 – Задання розрядності вхідного повідомлення

Код на вході k	Розрядність вхідного повідомлення, біт
001	16
010	24
011	32
100	40
101	48

Роботу кодера описано на мові Verilog. Моделювання та синтез кодера виконано в середовищі Quartus II (рисунок 4.8). Пристрій реалізовано на ПЛІС фірми «Altera», серія Cyclone IV (рисунок 4.9). Функціональна схема кодера приведена на рисунку 4.11.

Для перевірки правильності роботи кодера вхідне повідомлення X та вихідні значення x_1 , x_2 , x_3 , x_4 на часовій діаграмі подані в двійковій системі числення, значення $A32$ отримане по залишках (x_1, x_2, x_3, x_4) і перевірочні символи x_5 і x_6 подані в десятковій системі числення (рисунку 4.9):

$$x_5 = A32(\text{mod } p5) = 2260348044(\text{mod } 277) = 67;$$

$$x_6 = A32(\text{mod } p6) = 2260348044(\text{mod } 281) = 61.$$

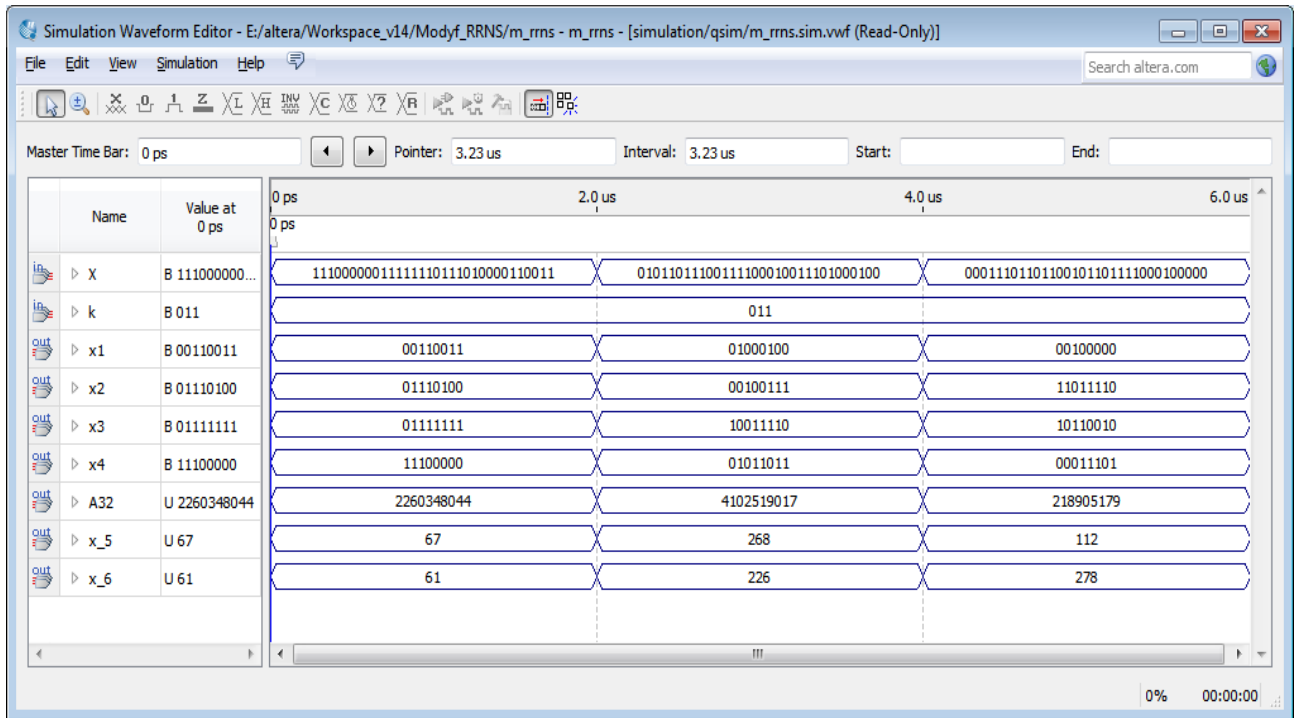


Рисунок 4.8 – Результати симуляції роботи кодера при розрядності вхідних даних $n=32$

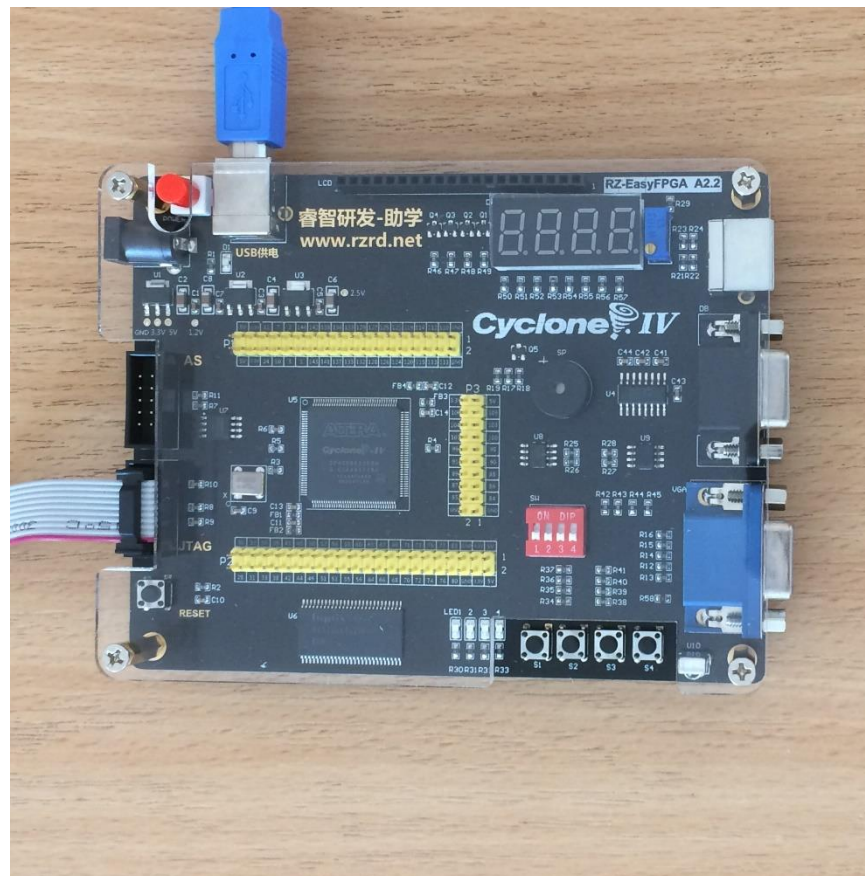


Рисунок 4.9 – Стенд для реалізації та верифікації роботи кодера на програмованій логічній інтегральній схемі Cyclone IV

Інтерфейс кодера наведено на рисунку 4.10. При $h=0$ – обчислюється один перевірочний символ, при $h=1$ – два перевірочні символи. Вхід k визначає розрядність вхідного повідомлення відповідно до таблиці 4.2.

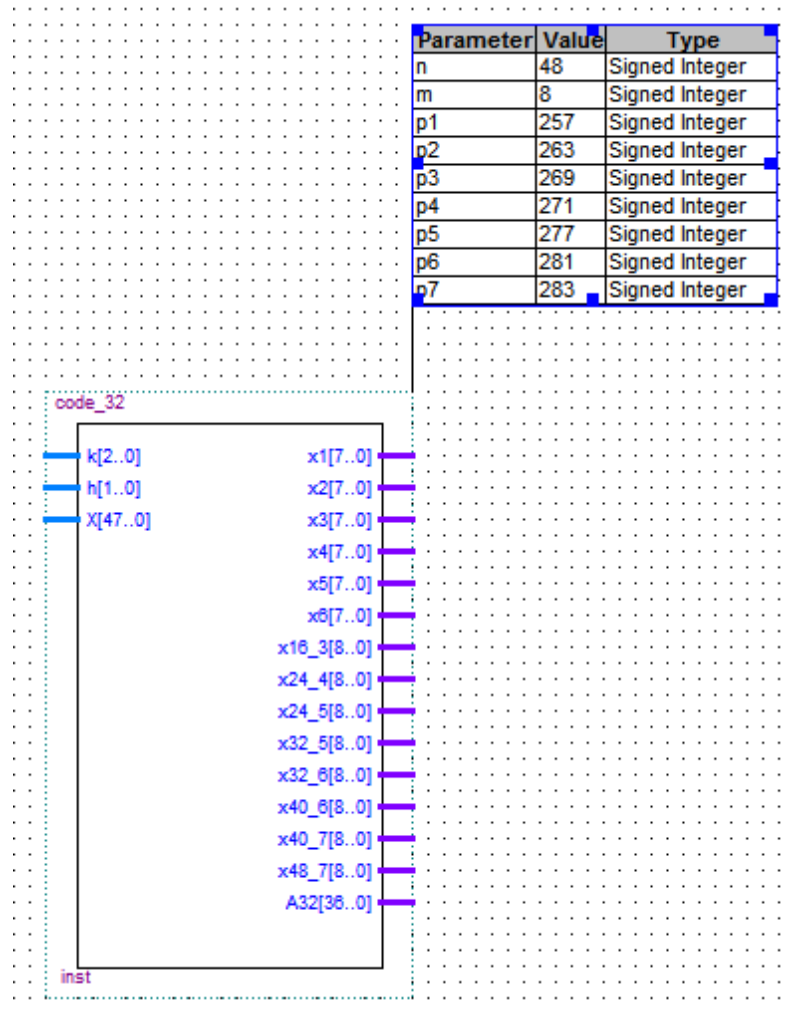


Рисунок 4.10 – Інтерфейс кодера

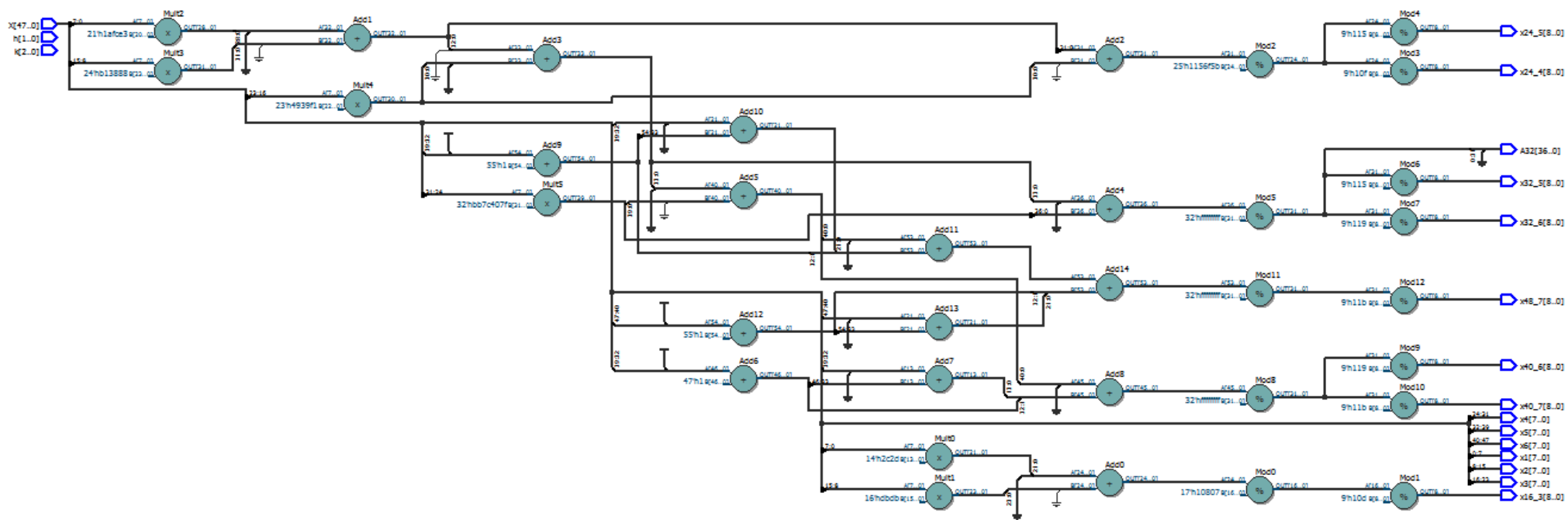


Рисунок 4.11 – Функціональна схема кодера

Апаратні затрати (кількість логічних елементів) та максимальна затримка сигналу при реалізації кодера на ПЛІС для різної розрядності вхідних даних наведені в таблиці 4.3.

Таблиця 4.3 – Параметри синтезу кодера для різної розрядності повідомлення

Розрядність повідомлення	16	24	32	40	48
Всього логічних елементів	1,113 / 21,280 (5 %)	1,846 / 21,280 (9 %)	2,472 / 21,280 (12 %)	3,127 / 21,280 (15 %)	3,745 / 21,280 (18 %)
Всього контактів	84 / 167 (50 %)	92 / 167 (55 %)	100 / 167 (60 %)	108 / 167 (65 %)	116 / 167 (69 %)
Час затримки, нс	71.663	101.822	118.132	148.625	187.592

Експериментально встановлено, що реалізація кодера на основі запропонованого методу формування коригуючих кодів СЗК порівняно з відомими методами забезпечує зменшення апаратних затрат на 20% залежно від розрядності повідомлення та підвищення швидкодії за рахунок відсутності перетворення повідомлення в СЗК.

4.3 Пристрій завадостійкого кодування/декодування даних на основі двовимірних модулярних коригуючих кодів

Реалізовано на ПЛІС кодер/декодер виявлення та виправлення пакетів помилок на основі модулярних коригуючих кодів та двовимірної схеми.

4.3.1 Реалізація кодера

Процес кодування відбувається наступним чином (рисунок 4.12). На вхід кодера поступає повідомлення з k символів. У кодері обчислюється перевірочний символ, і повідомлення з перевірочним символом записується в буфер. Процес повторюється доти, поки не буде прийнято k повідомлень. З прийнятих повідомлень формується масив даних, з якого обчислюються перевірочні символи по стовпцях масиву.

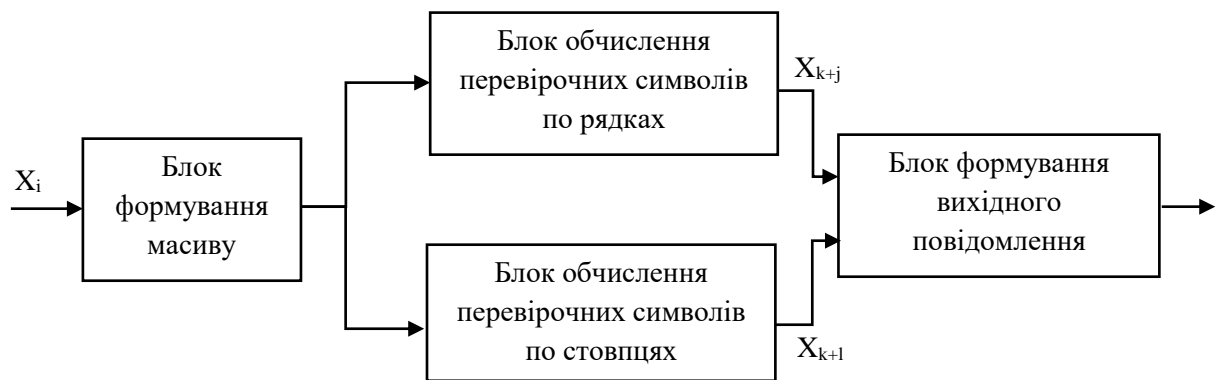


Рисунок 4.12 – Структура кодера (X_i – вхідне повідомлення)

Після того як у кодері завершиться обчислення перевірочних символів по стовпцях, на виході формується новий масив прийнятого повідомлення з обчисленими перевірочними символами по рядках і по стовпцях. Структура повідомлення на виході кодера подана на рисунку 4.13.

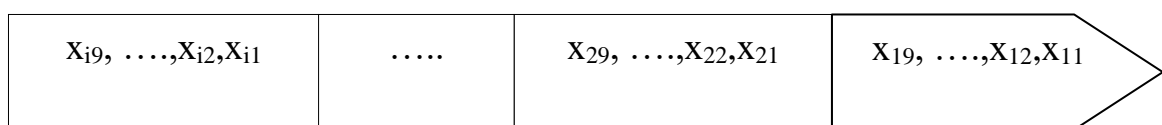


Рисунок 4.13 – Структура вихідного повідомлення

Приклад завадостійкого кодування на основі модулярних кодів та двовимірної схеми. Вхідні повідомлення, сформовані у вигляді двовимірного масиву, наведені в таблиці 4.4.

Для реалізації кодера виберемо взаємно прості коефіцієнти v_i :
 $v_1 = 211, v_2 = 257, v_3 = 263, v_4 = 269, v_5 = 271, v_6 = 277, v_7 = 283,$
 $v_8 = 288$ та модуль $P = 131297$.

Таблиця 4.4 – Подання повідомлення у вигляді масиву

	x1	x2	x3	x4	x5	x6	x7	x8
y1	10	4	3	4	8	5	4	10
y2	9	8	4	9	7	3	8	8
y3	9	1	3	8	7	3	6	10
y4	4	4	10	7	2	9	10	6
y5	5	2	9	2	3	1	4	4
y6	3	9	9	8	8	9	2	8
y7	2	6	6	2	7	4	6	6
y8	8	4	4	1	10	5	6	7

Обчислюємо перевірочні символи по рядках за формулою (2.13)
(таблиця 4.5).

Таблиця 4.5 – Повідомлення з обчисленими перевірочними символами
по рядках

	x1	x2	x3	x4	x5	x6	x7	x8	x_{k+j}
x1	10	4	3	4	8	5	4	10	10128
x2	9	8	4	9	7	3	8	8	14392
x3	9	1	3	8	7	3	6	10	12361
x4	4	4	10	7	2	9	10	6	13988
x5	5	2	9	2	3	1	4	4	8130
x6	3	9	9	8	8	9	2	8	15512
x7	2	6	6	2	7	4	6	6	11037
x8	8	4	4	1	10	5	6	7	12960

Перевірочні символи, обчислені по стовпцях приведені в таблиці 4.6.

Таблиця 4.6 – Повідомлення з обчисленими перевірочними символами по стовбцях

	x1	x2	x3	x4	x5	x6	x7	x8
x1	10	4	3	4	8	5	4	10
x2	9	8	4	9	7	3	8	8
x3	9	1	3	8	7	3	6	10
x4	4	4	10	7	2	9	10	6
x5	5	2	9	2	3	1	4	4
x6	3	9	9	8	8	9	2	8
x7	2	6	6	2	7	4	6	6
x8	8	4	4	1	10	5	6	7
x_{k+1}	12922	10124	12922	10756	13756	10372	12232	15424

Роботу кодера описано на мові Verilog та синтезовано на ПЛІС фірми Altera. Функціональна схема кодера подана на рисунку 4.14.

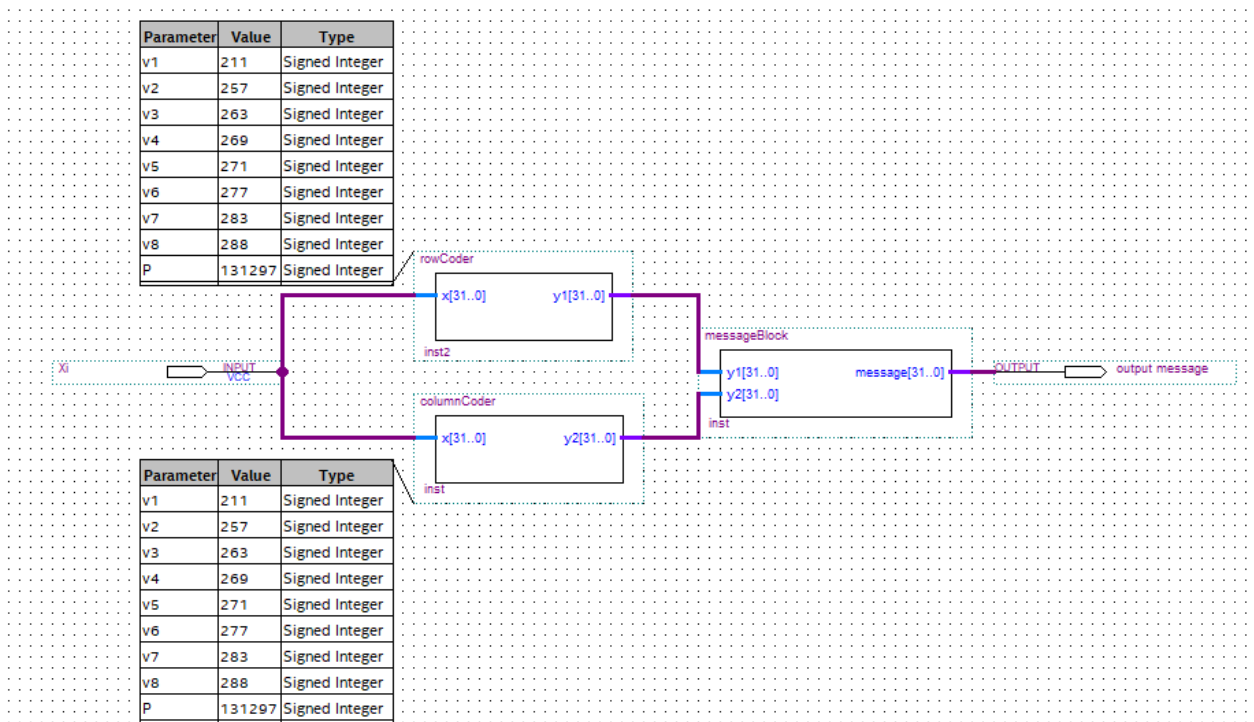


Рисунок 4.14 – Функціональна схема кодера

4.3.2 Реалізація декодера

Декодер по прийнятих даних обчислює перевірочні символи, та порівнює їх з тими, які були обчислені кодером та надіслані.

Для виправлення помилок у пакеті даних у декодері формується таблиця синдромів по рядках та стовбцях.

Виправлення однократних помилок відбувається на основі порівняння обчисленого синдрому з таблицею синдромів. Якщо значення співпадають – помилка виявлена і її можна виправити. Після цього обчислюється таблиця синдромів по рядках і стовбцях, якщо значення синдрому дорівнює нулю, то помилки були виправлені.

Приклад функції для виявлення та виправлення однократних помилок у пакеті даних має вигляд:

```
def search_mistake(mass,mass2):
    Error_first_index = -1
    Error_last_index = -1
    for i in range(size):
        if (mass[i][size]-mass2[i][size]!=0):
            Error_first_index = i
        if (mass[size][i]-mass2[size][i]!=0):
            Error_last_index = i
    Error = [Error_first_index,Error_last_index]
    return Error
tmp = search_mistake(mass,mass2)
first = int(tmp[0])
last = int(tmp[1])
print(first,last)
```

Якщо значення в таблиці синдромів не дорівнюють нулю, то це означає, що помилки виявлені і вони розміщені в двох рядках. Для виправлення таких помилок необхідно розв'язати рівняння (2.16) для

кожного рядка/стовбця (блок виправлення багатократних помилок) (див. рисунок 4.15).

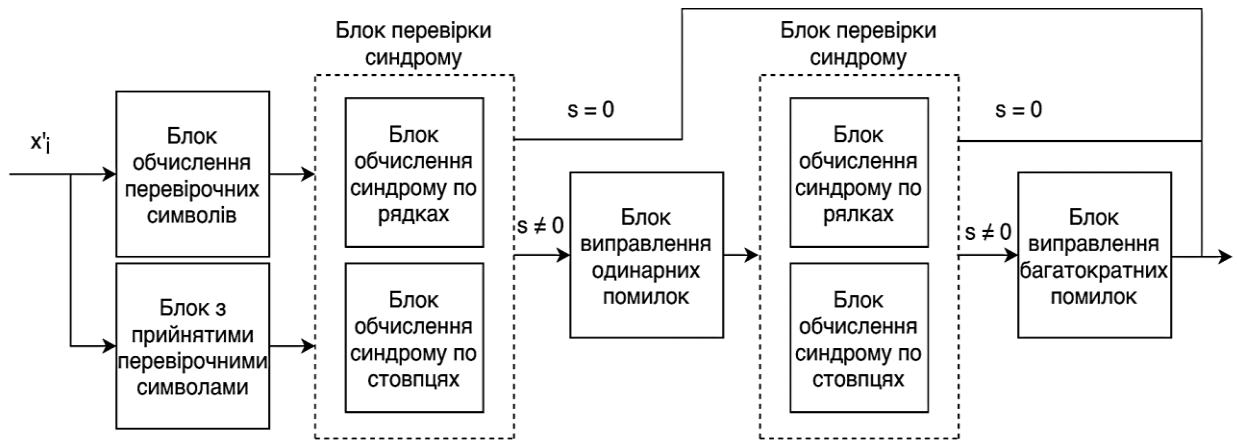


Рисунок 4.15 – Структура декодера

Розроблений пристрій завадостійкого декодування даних на основі двовимірних модулярних коригуючих кодів дає змогу виявляти та виправляти велику кількість помилок у пакеті даних.

Програмна реалізація декодера для виявлення та виправлення помилок у пакеті даних розроблена на мові програмування Python (Додаток Б).

4.4 Адаптивний протокол передачі даних на основі модулярних коригуючих кодів

На основі запропонованого в пункті 2.2 методу формування коригуючих кодів системи залишкових класів розроблено адаптивний алгоритм

(рисунок 4.16) та протокол передачі даних у безпроводних сенсорних мережах [59, 114].

Суть запропонованого адаптивного алгоритму передачі полягає в тому, що пакет даних відправляється з додаванням одного перевірного символу,

який дозволяє тільки виявляти помилки в даних. Якщо помилку виявлено, то відправляється запит на передачу додаткового перевірного символу.

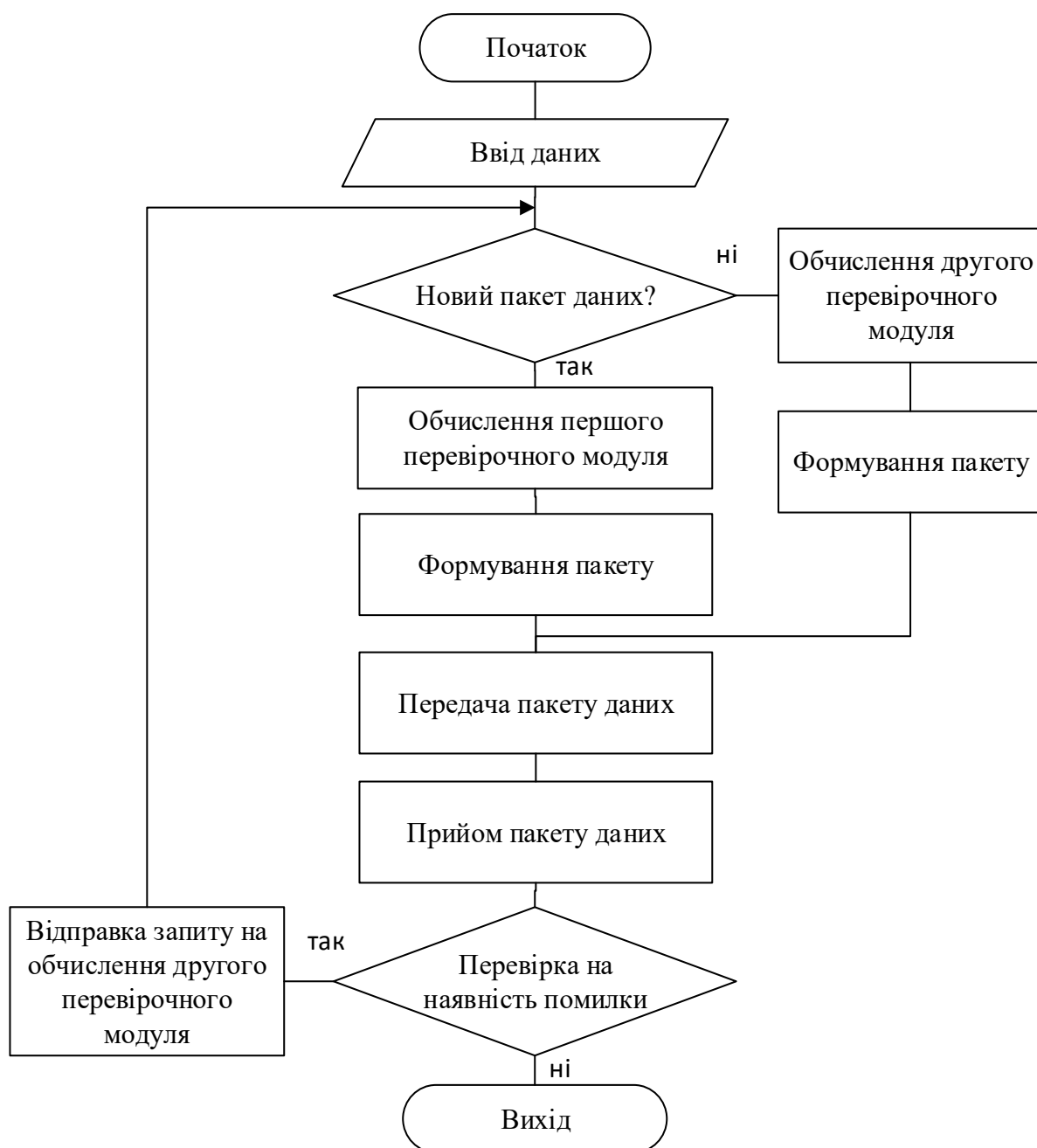


Рисунок 4.16 – Адаптивний алгоритм передачі даних

Для проведення експериментальних досліджень використано радіомодуль NRF24L01. Структура протоколу даного модуля наведена на рисунку 4.17.

Структура пакету складається з:

- преамбули – послідовність бітів (1 байт), які використовуються для синхронізації приймача і передавача;
- адреси отримувача – вказує адресу потрібного приймача. Довжина адреси приймача займає від 3 до 5 байт;
- контрольного поля – визначає довжину пакету даних (від 0 до 32 байт) та 2 біта PID для виявлення, чи є пакет даних новим, чи повторно надісланим, і 1 біт прапорця NO_ACK;
- блоку даних – довжина блоку даних від 0 до 32 байта (довжина поля даних може змінюватись);
- CRC – механізму виявлення помилок у пакеті даних (довжина CRC коду може бути від 1 до 2 байта).

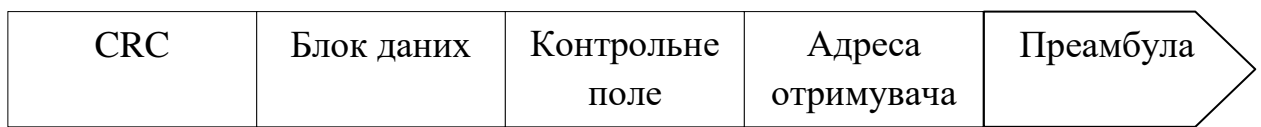
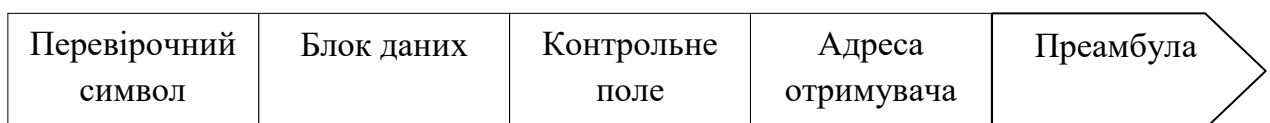


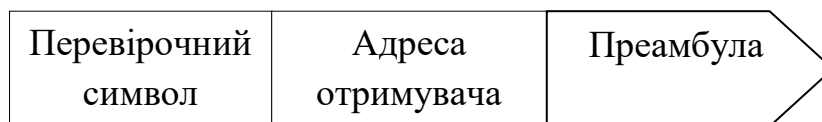
Рисунок 4.17 – Структурна схема протоколу

Структура протоколу з використанням коригуючих кодів системи залишкових класів наведена на рисунку 4.18 а.

Якщо помилка виявлена, то приймач відправляє запит на передачу наступного перевірного символу (рисунок 4.18 б). Наявність двох перевірих символів дозволяє виправляти помилки в одному інформаційному символі.



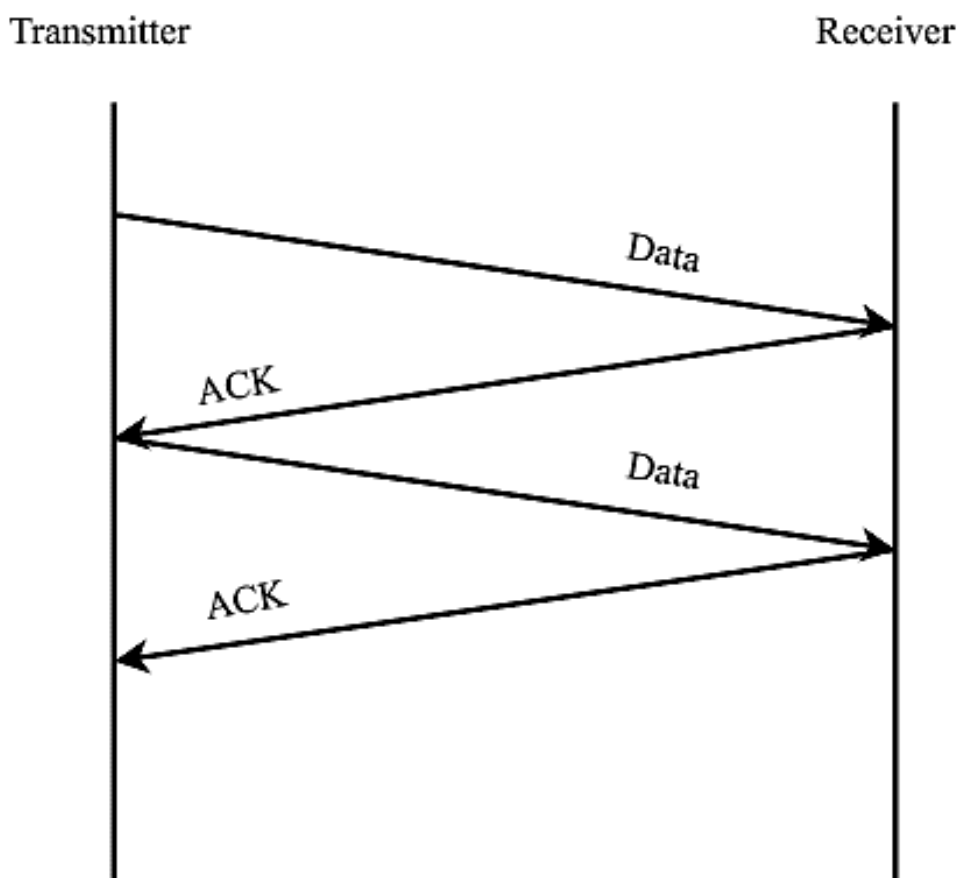
а)



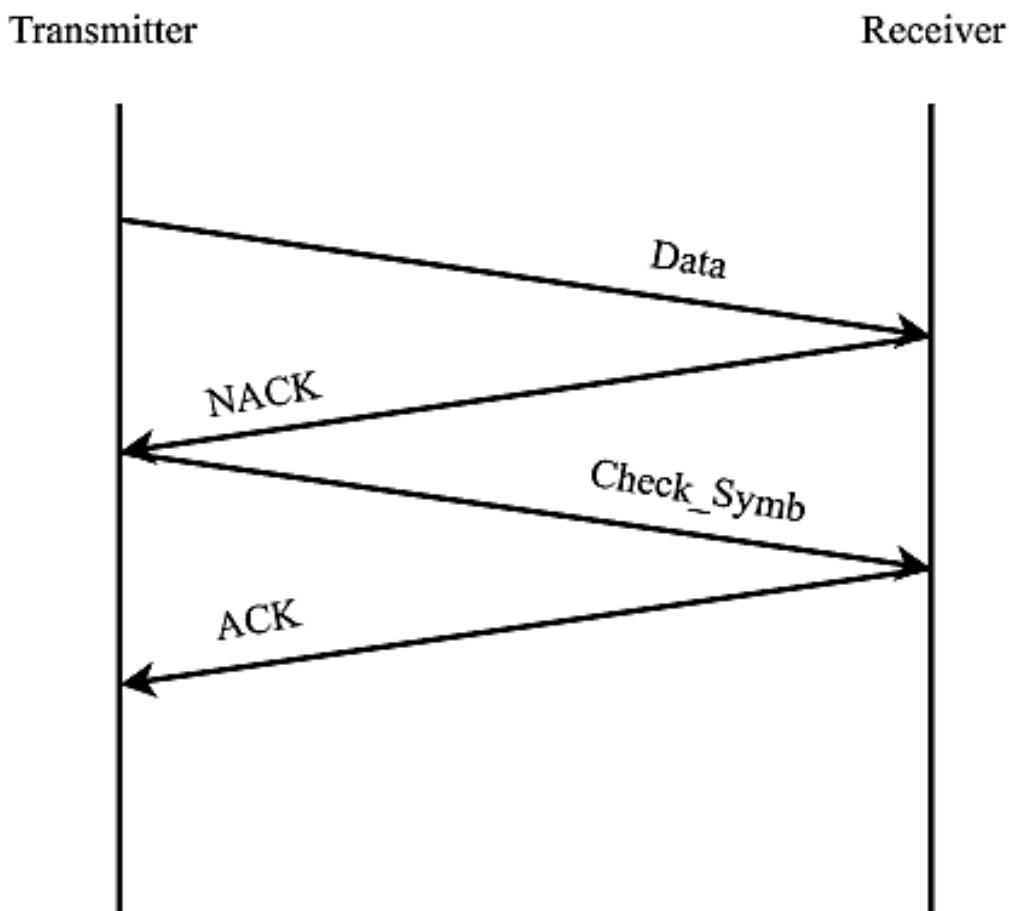
б)

Рисунок 4.18 – Удосконалена структура протоколу передачі даних

Протокол передачі гарантує надійних обмін даними за рахунок зворотного зв'язку у вигляді відповіді підтвердження ACK (рисунок 4.19 а). Якщо дані отримані з помилкою, відправляється пакет у вигляді відповіді NACK, і передавач відправляє додатковий перевірочний символ (рисунок 4.19 б).



а)



б)

Рисунок 4.19 – Послідовність обміну даними між передавачем і приймачем

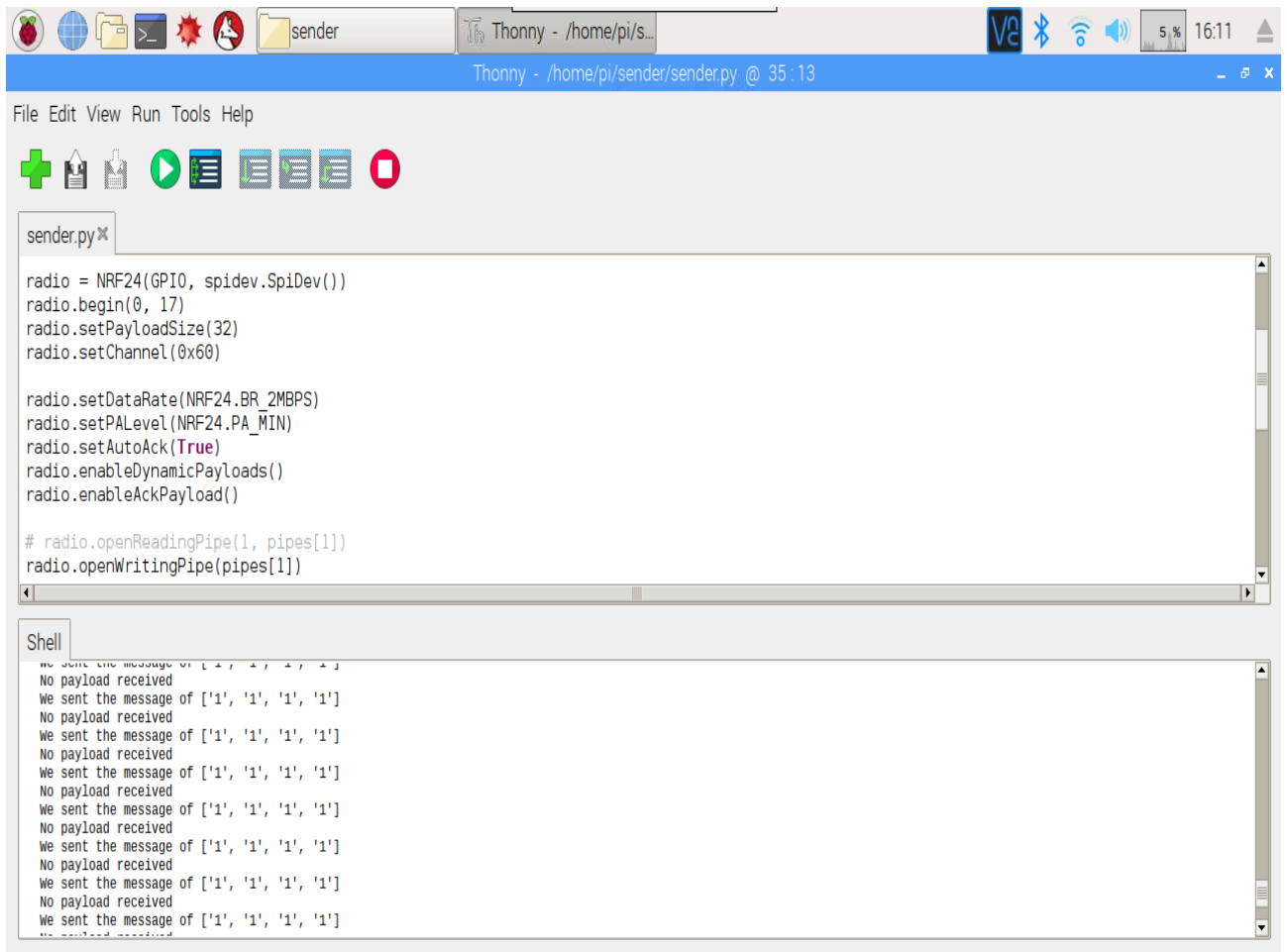
Розроблений протокол забезпечує зменшення надлишковості на 34% при виникненні одиничних символних помилок порівняно з протоколом Enhanced Shock Burst™ transceiver NRF24L01+. (Оскільки обсяг даних при повторній передачі пакету дорівнює 764 біт, а при передачі додаткового перевірного символу – 502 біти).

Програмний код адаптивного алгоритму для передавача та приймача на основі модулярних коригуючих кодів реалізований на мові програмування Python.

Для налаштування радіомодуля NRF24L01 використана бібліотека NRF24, яка дозволяє використовувати радіомодуль одночасно як передавач, так і приймач.

Бібліотека NRF24 дозволяє вибирати такі параметри як: радіочастотний канал зв'язку (0-127); кількість байтів даних, які будуть передаватись (по замовчуванню – 32 байт) та інше.

На рисунку 4.20 зображено вікно програми передавача, який передає текстові повідомлення довжиною чотири байти.



The screenshot shows the Thonny IDE interface. The top bar indicates the file path as 'Thonny - /home/pi/sender/sender.py @ 35:13'. The main editor area contains the following Python code:

```

radio = NRF24(GPIO, spidev.SpiDev())
radio.begin(0, 17)
radio.setPayloadSize(32)
radio.setChannel(0x60)

radio.setDataRate(NRF24.BR_2MBPS)
radio.setPALevel(NRF24.PA_MIN)
radio.setAutoAck(True)
radio.enableDynamicPayloads()
radio.enableAckPayload()

# radio.openReadingPipe(1, pipes[1])
radio.openWritingPipe(pipes[1])

```

Below the code editor is a 'Shell' window showing the program's output:

```

We sent the message of ['1', '1', '1', '1']
No payload received
We sent the message of ['1', '1', '1', '1']
No payload received
We sent the message of ['1', '1', '1', '1']
No payload received
We sent the message of ['1', '1', '1', '1']
No payload received
We sent the message of ['1', '1', '1', '1']
No payload received
We sent the message of ['1', '1', '1', '1']
No payload received
We sent the message of ['1', '1', '1', '1']
No payload received
We sent the message of ['1', '1', '1', '1']
No payload received
We sent the message of ['1', '1', '1', '1']
No payload received
We sent the message of ['1', '1', '1', '1']
No payload received
We sent the message of ['1', '1', '1', '1']

```

Рисунок 4.20 – Вікно передавача

На рисунку 4.21 зображено вікно приймача, який приймає повідомлення та перевіряє останнє на наявність помилки. Якщо прийняте повідомлення з помилкою, тоді програма зберігає інформацію про помилку в окремому файлі.

The screenshot shows the Thonny IDE interface. The top bar indicates the file path is `/home/pi/recv-1.py` and the current time is 16:12. The menu bar includes File, Edit, View, Run, Tools, and Help. Below the menu bar are icons for file operations and execution. The main editor window displays the following Python code:

```

recv-1.py
import RPi.GPIO as GPIO
from lib_nrf24 import NRF24
import time
import spidev

GPIO.setmode(GPIO.BCM)

pipes = [[0xe7, 0xe7, 0xe7, 0xe7, 0xe7], [0xc2, 0xc2, 0xc2, 0xc2, 0xc2]]

radio = NRF24(GPIO, spidev.SpiDev())
radio.begin(0, 17)
radio.setPayloadSize(64)
radio.setChannel(0x60)

```

The Shell window at the bottom shows the execution output:

```

Received: [49, 49, 49, 49]
Translating the receivedMessage into unicode characters
1111
Received: [49, 49, 49, 49]
Translating the receivedMessage into unicode characters
1111
Received: [49, 49, 49, 49]
Translating the receivedMessage into unicode characters
1111
Received: [49, 49, 49, 49]
Translating the receivedMessage into unicode characters
1111
Received: [49, 49, 49, 49]
Translating the receivedMessage into unicode characters
1111

```

Рисунок 4.21 – Вікно приймача

Прийняті пакети повідомлень записуються в файл для подальшого опрацювання. При виявленні помилки повідомлення записується в окремий файл.

Для того, щоб визначити структуру помилки та вибрати необхідні параметри коригуючого коду в бібліотеці NRF24, було відключено перевірку виявлення помилок (CRC код).

4.3.1 Підключення радіомодуля NRF24L01

Для проведення експериментальних досліджень адаптивного протоколу передачі даних використано радіомодулі NRF24L01 та одноплатний комп'ютер Raspberry Pi.

Радіомодуль NRF24L01 працює на частоті 2.4 ГГц зі швидкістю передавання до 2 Мбіт / с. Керування модулем здійснюється по інтерфейсу SPI [57]. Радіомодуль використовує один з 126 радіочастотних каналів та може працювати в режимі один до багатьох. Радіомодуль підтримує апаратний контроль помилок з використання CRC кодів, контроль адресації при організації зв'язку один до багатьох. Даний радіомодуль має вбудований регулятор напруги від 1.9 В до 3.6 В, що дозволяє споживати 1мкА в режимі економії енергії (Power down). Радіомодуль NRF24L01 має розміри 15 x 29 мм і використовується в промислових системах управління (рисунок 4.22).



Рисунок 4.22 – Радіомодуль NRF24L01

При проведенні експериментальних досліджень розробленого адаптивного протоколу використано одноплатний комп'ютер Raspberry Pi 3B, до якого приєднаний датчик вологості та радіомодуль NRF24L01, який виконує роль передавача (рисунок 4.23).

Перевагою вибраного рішення на основі Raspberry Pi 3B є невеликі розміри та висока обчислювальна потужність, а саме: чотирьох-ядерний процесор Broadcom з тактовою частотою 1.2 ГГц із 64-розрядною шиною; 1 ГБ вбудованої оперативної пам'яті; 40 входів/виходів для підключення додаткових модулів та сенсорів [58].

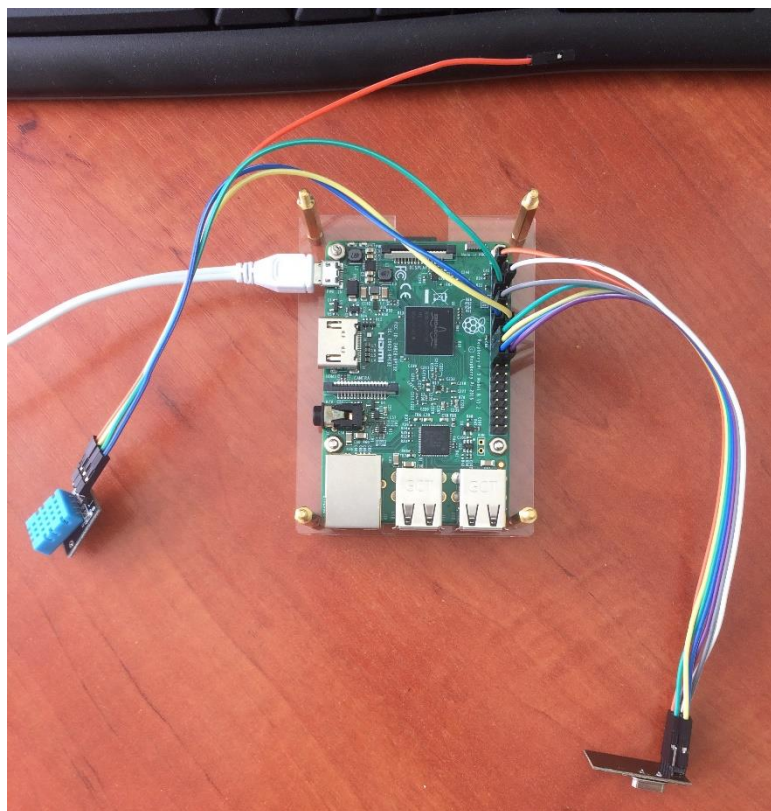


Рисунок 4.23 – Одноплатний комп'ютер Raspberry Pi 3B

У таблиці 4.7 наведені назви виводів радіомодуля NRF24L01 та Raspberry Pi 3B для організації зв'язку між двома одноплатними комп'ютерами.

Таблиця 4.7 – Послідовність підключення радіомодуля NRF24L01 до Raspberry Pi 3B

Радіомодуль NRF24L01	Raspberry Pi 3B (BCM позначення)	Raspberry Pi 3B (позначення на платі)
Vcc	3.3v	Pin 1
GND	Ground	Pin 6
CSN	GPIO08	Pin 24
CE	GPIO17	Pin 11
MOSI	GPIO10	Pin 19
MISO	GPIO09	Pin 21
SCK	GPIO11	Pin 23

Підключення радіомодуля NRF24L01 до одноплатного комп'ютера Raspberry Pi 3В здійснюється по інтерфейсу SPI (рисунок 4.24).

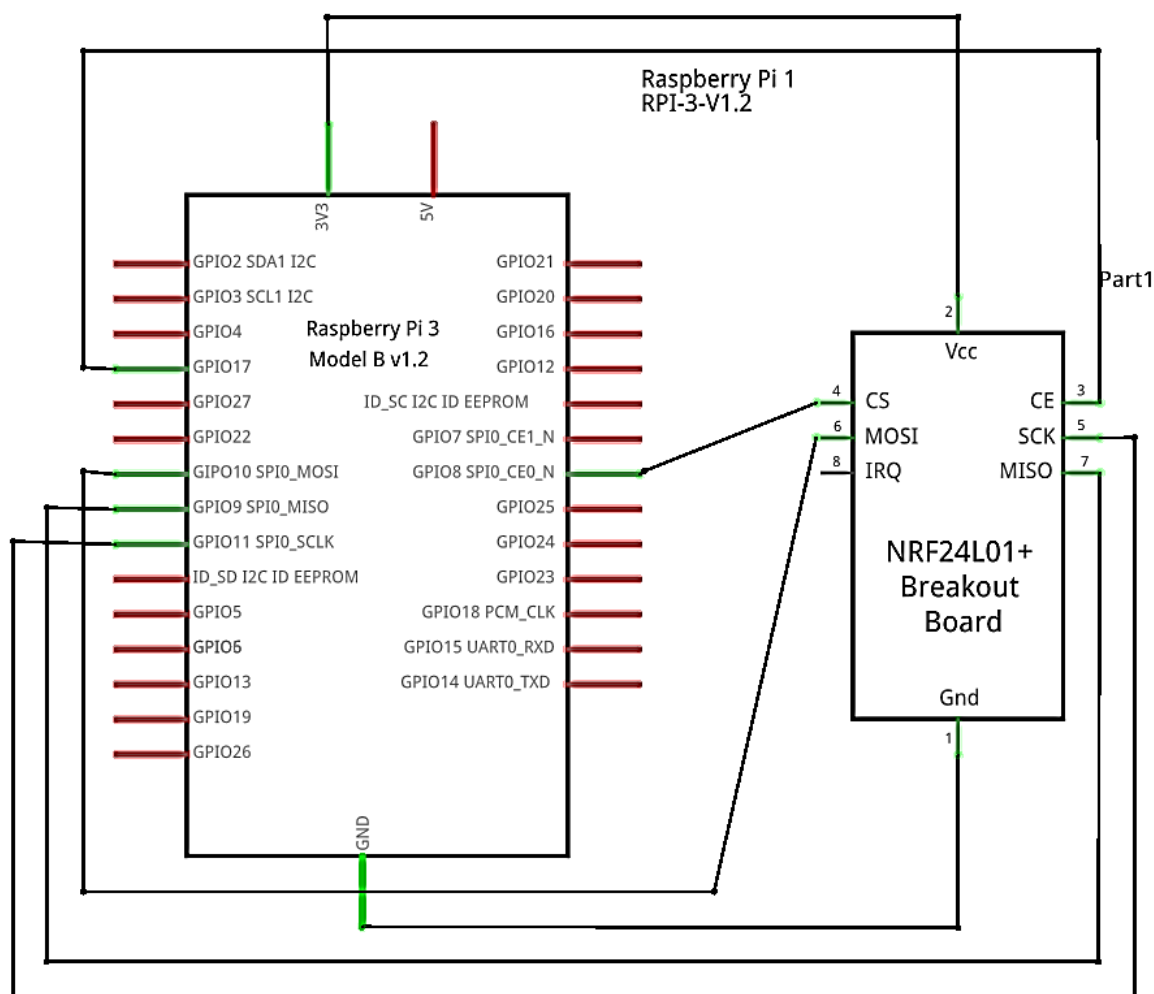


Рисунок 4.24 – Схема підключення радіомодуля NRF24L01 до Raspberry Pi 3В

Проведено дослідження тривалості передачі пакетів різної довжини при використанні алгоритму з повторною передачею спотворених пакетів (алгор.1) та запропонованого адаптивного алгоритму (алгор.2) (рисунок 4.25).

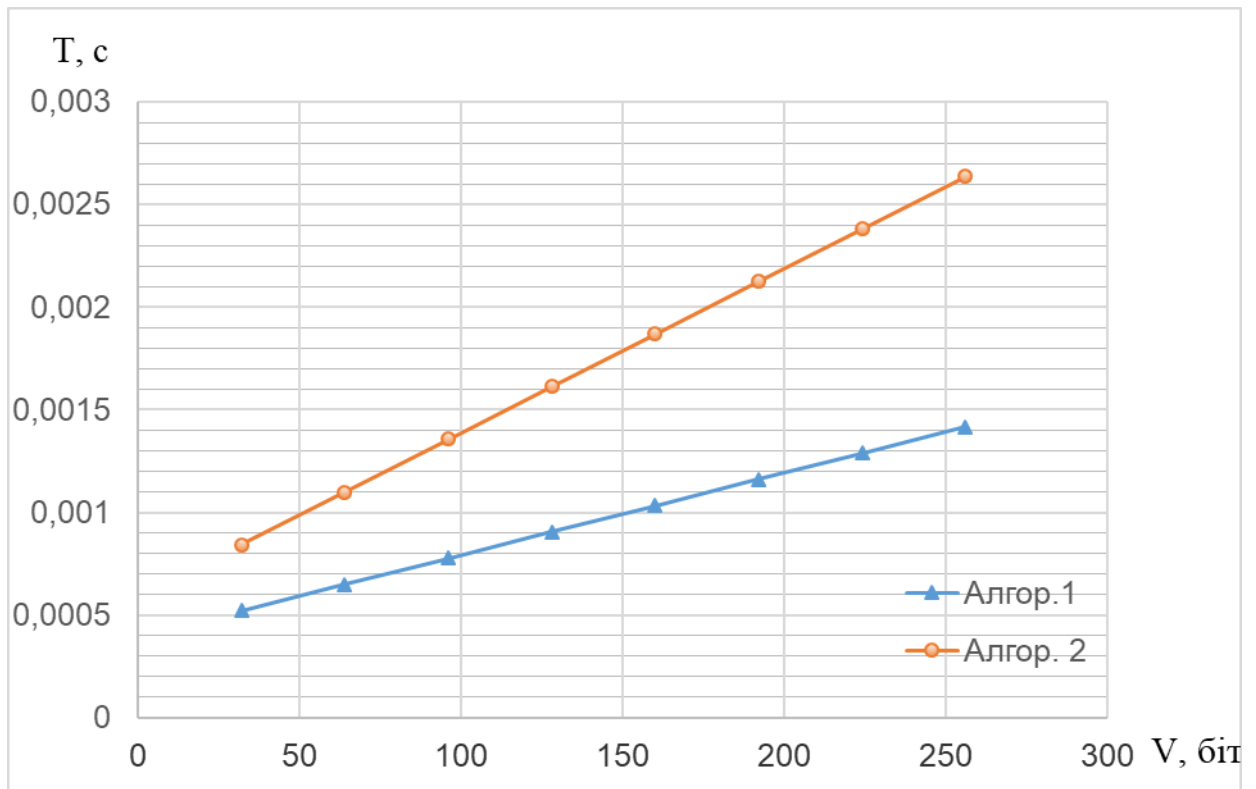


Рисунок 4.25 – Порівняльний аналіз відомого та запропонованого алгоритму

Адаптивний алгоритм передачі даних у безпроводних сенсорних мережах підвищує надійність передачі та зменшує надлишковість при використанні коригуючих кодів системи залишкових класів. Як видно з рисунку 4.27, запропонований алгоритм забезпечує зменшення часу на повторну передачу спотворених пакетів у середньому на 80 %.

Висновки до четвертого розділу

1. Розроблено пристрій виправлення помилок на основі коригуючих кодів системи залишкових класів зі спеціальною системою модулів на основі методів обчислення проєкцій числа та обчислення синдрому. Проведені дослідження показали, що апаратні затрати при реалізації на ПЛІС алгоритму

виправлення помилок на основі методу обчислення проєкцій є меншими приблизно в 3,6 рази порівняно з методом обчислення синдрому.

2. Реалізовано на ПЛІС кодер коригуючих кодів системи залишкових класів. Експериментально встановлено, що реалізація кодера на основі запропонованого методу формування коригуючих кодів СЗК порівняно з відомими методами забезпечує зменшення апаратних затрат на 20% залежно від розрядності повідомлення та підвищення швидкодії за рахунок відсутності перетворення повідомлення в СЗК.

3. Розроблено адаптивний алгоритм та протокол передачі даних у безпроводних сенсорних мережах на основі коригуючих кодів СЗК. Проведені експериментальні дослідження показали, що адаптивний алгоритм передачі даних забезпечує зменшення часу на повторну передачу спотворених пакетів у середньому на 80%.

4. Розроблено структуру та реалізовано на ПЛІС кодер для обчислення перевірочних символів і формування пакету даних на основі двовимірних модулярних коригуючих кодів. Декодер виявлення та виправлення пакетів помилок на основі модулярного коригуючого коду реалізовано програмно на мові Python.

ВИСНОВКИ

У дисертаційній роботі розв'язано актуальну науково-прикладну задачу підвищення надійності передачі даних у безпроводних сенсорних мережах шляхом розробки методів та засобів завадостійкого кодування даних на основі модулярної арифметики. Отримано наступні наукові результати.

1. Проведено аналіз методів та алгоритмів, які використовуються для підвищення надійності передачі даних у безпроводних сенсорних мережах. Встановлено, що відомі методи, не забезпечують необхідну надійність передачі даних або мають високу надлишковість.

2. Удосконалено коригуючі коди в системі залишкових класів за рахунок вибору спеціальної системи модулів. Проведені дослідження операції обчислення залишку з використанням довільної та спеціальної системи модулів засвідчили, що застосування системи спеціальних модулів забезпечує зменшення апаратних затрат у 2,8 рази та збільшення швидкодії виконання перетворення на 29 %.

3. Запропоновано метод формування перевірочних символів у коригуючих кодах системи залишкових класів, який забезпечує зменшення апаратних затрат в середньому на 26 % залежно від розрядності повідомлення та підвищення швидкодії за рахунок відсутності процедури перетворення повідомлення з позиційної системи в систему залишкових класів і зворотного перетворення.

4. Розроблено метод виправлення помилок на основі модулярних коригуючих кодів, який забезпечує виправлення помилок у двох інформаційних символах з використанням одного перевірочного символу, що дозволяє збільшити швидкість коду приблизно на 20 %, а надлишковість модулярного коригуючого коду, відповідно, зменшити.

5. Розроблено двовимірний метод контролю помилок на основі модулярних коригуючих кодів, який забезпечує ефективне виправлення

пакетів помилок у безпроводних каналах зв'язку, характеризується високою коригуючою здатністю та низькою складністю реалізації алгоритму декодування і, відповідно, може бути застосований для підвищення надійності передачі даних у безпроводних сенсорних мережах.

6. Розроблено пристрій виправлення помилок на основі коригуючих кодів системи залишкових класів зі спеціальною системою модулів на основі методів обчислення проєкцій числа та обчислення синдрому. Проведені дослідження показали, що при реалізації на ПЛІС декодера виправлення помилок на основі спеціальної системи модулів забезпечує зменшення апаратних затрат приблизно на 26%, що становить 6,5 тис. логічних елементів для методу проєкцій та приблизно 10 % для методу обчислення синдрому. Використання спеціальної системи модулів підвищує швидкість роботи декодера приблизно на 18 % для методу обчислення проєкцій та на 9 % для методу обчислення синдрому.

7. Реалізовано на ПЛІС кодер обчислення для визначення перевіірочних символів у коригуючих кодах системи залишкових класів. Експериментально встановлено, що реалізація кодера на основі запропонованого методу формування коригуючих кодів СЗК порівняно з відомими методами забезпечує зменшення апаратних затрат на 26% і залежить від розрядності повідомлення.

8. Розроблено адаптивний алгоритм та протокол передачі даних у безпроводних сенсорних мережах на основі коригуючих кодів системи залишкових класів. Проведені експериментальні дослідження показали, що адаптивний алгоритм передачі даних забезпечує зменшення часу на повторну передачу спотворених пакетів у середньому на 80 %.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Agre J., Clare L.. An integrated architecture for cooperative sensing networks, *IEEE Computer Magazine* (May 2000) P. 106–108.
2. Akyildiz F., Pompili D., Melodia T. Underwater Acoustic Sensor Networks: Research Challenges, *Ad Hoc Net.*, vol. 3, no. 3, May 2005. P. 257–79.
3. Akyildiz I.F, Stuntebeck E.P. Wireless underground sensor networks: Research challenges, *Ad Hoc Networks Journal*, 2006. No4. P. 669–686.
4. Akyildiz I.F., Melodia T., Chowdury K.R. Wireless multimedia sensor networks: applications and testbeds. *Proceedings of the IEEE*. 2008. Vol. 96. No 10. P. 1588-1605.
5. Akyildiz I.F., Vuran M.C. *Wireless Sensor Networks*, New York: John Wiley & Sons, 2010. P. 571.
6. Akyildiz Ian F., Zhi Sun, Mehmet C. Vuran. Signal propagation techniques for wireless underground communication networks. *Physical Communication*, 2009. No2. P. 167–183.
7. Alia G. and Martinelli E., “A VLSI modulo m multiplier,” *IEEE Trans. on Computers*, Vol. C-40, No. 7, July 1991. P. 873-878.
8. Alrajeh, N.A., Marwat, U., Shams, B., & Shah, S.S.H. Error correcting codes in wireless sensor networks: an energy perspective. *Applied Mathematics & Information Sciences*, 2015, P. 809-818.
9. Alves Jr. J., Nascimento L.F.L., Albin L.C.P.. Using the redundant residue number system to increase routing dependability on mobile ad hoc networks. *Cyber Journals: J. of Selected Areas in Telecommunications* 2(1), P. 67–73 (2011)
10. Antoniou P., Pitsillides A., Blackwell T., Engelbrecht A., Michael L. “Congestion Control in Wireless Sensor Networks based on Bird Flocking Behavior,” *Computer Networks*, Vol. 57, Issue 5. 2013. P. 1167–1191.

11. Avik S., Balasubramaniam N.. Performance of systematic RRNS based space-time block codes with probability-aware adaptive demapping. *IEEE Trans. Wirel. Commun.* 12(5), 2013. P. 2458–2469
12. Balen J., Zagar D., Martinovic G. Quality of service in wireless sensor networks: a survey and related patents // *Recent Patents on Computer Science*. 2011. V.4. P. 188–202.
13. Barsi F., Maestrini P. Error correcting properties of redundant residue number systems. *IEEE Trans. Comput.* 1973. C-22. P. 307–315.
14. Behren R, Condit J, Brewer E. Why Events are a Bad Idea (for High Concurrency Servers). *Proceedings of the 9th Workshop on Hot Topic in Operating Systems (HOTOS IX)*; Lihue, HI, USA. 18–21 May 2003. P. 19-24.
15. Bhatti S, Carlson J, Dai H, Deng J, Rose J, Sheth A, Shucker B, Gruenwald C, Torgerson HR. Mantis OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms. *Mob. Netw. Appl.* 2005. P. 563–579.
16. Bulusu N., Estrin D., Girod L., Heidemann J.. Scalable coordination for wireless sensor networks: self-configuring localization systems. *International Symposium on Communication Theory and Applications (ISCTA 2001)*, Ambleside, UK, July 2001.
17. Cao Q, Abdelzaher T, Stankovic J, He T. The LiteOS Operating System: Towards Unix Like Abstraction for Wireless Sensor Networks. *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008)*; St Louis, MO, USA. 22–24 April 2008. P. 233-244.
18. Cerpa A., Elson J., Hamilton M., Zhao J.. Habitat monitoring: application driver for wireless communications technology. *ACM SIGCOMM'2000*, Costa Rica, April 2001. P. 20-41.
19. Chen, D., Varshney, P. K. QoS Support in Wireless Sensor Networks: A Survey. In *International conference on wireless networks*. 2004. Vol. 233, P. 1-7.

20. Chien C., Elgorriaga I., McConaghy C.. Low-power direct-sequence spread-spectrum modem architecture for distributed wireless sensor networks. ISLPED'01, Huntington Beach, California, August 2001. P. 251-254.
21. Cho S., Chandrakasan A.. Energy-efficient protocols for low duty cycle wireless microsensor, Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI Vol. 2 (2000), P. 10.
22. De S., Qiao C., Wu H. Meshed Multipath Routing: An Efficient Strategy in Sensor Network. Computer Networks, Special Issue on Wireless Sensor Networks, Vol. 43, Issue 4, Nov, 2003. P. 481-497.
23. Dulman S., Havinga P, Nieberg T. Multipath Routing with Erasure Coding for Wireless Sensor Networks // In: ProRISC 2004, 15th Annual Workshop on Circuits, Systems and Signal Processing, 25-26 Nov. 2004, Veldhoven, the Netherlands. P. 181-188.
24. Dunkels A, Gronvall B, Voigt T. Contiki a Lightweight and Flexible Operating System for Tiny Networked Sensors. Proceedings of the 9th Annual IEEE International Conference on Local Computer Networks; Washington, DC, USA. October 2004; P. 455–462.
25. Dwivedi A.K, Tiwari M.K, Vyas O.P. Operating Systems for Tiny Networked Sensors: A Survey. IJRTE. 2009. P. 152–157.
26. Etzel M., Jenkins W.K. Redundant residue number systems for error detection and correction in digital filters. IEEE Trans. Acoust., Speech, Signal Process. 28 (5), 1980. P.538–545.
27. Gay D, Levis P, von Behren R, Welsh M, Brewer E, Culler D. The NesC Language: A Holistic Approach to Networked Embedded Systems. Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation; New York, NY, USA. May 2003. P. 41-51.
28. Ghune P., Ghune R. N., Pandey P., Mishra P. “Application of Wireless Sensor Networks in Smart Grid Opportunities, Challenges & Technologies Available (A Survey)”, International Journal on Advanced Computer Theory and Engineering (IJACTE), Vol. 2, Issue. 5. 2013. P. 11-18.

29. Goh, V. T., & Siddiqi, M. U. Multiple error detection and correction based on redundant residue number systems. *IEEE Transactions on Communications*, 56(3). 2008. P. 325-330.
30. Halweil B.. Study finds modern farming is costly, *World Watch* 14(1) (2001). P. 9–10.
31. Haron N.Z., Hamdioui S.. Redundant residue number system code for fault-tolerant hybrid memories. *ACM J. Emerg. Technol. Comput. Syst.* 7(1), 4. 2011.
32. Heinzelman W.R., Kulik J., Balakrishnan H.. Adaptive protocols for information dissemination in wireless sensor networks, *Proceedings of the ACM MobiCom'99*, Seattle, Washington, 1999. P. 174–185.
33. Himani Chawla. “Some issues and challenges of Wireless Sensor Networks,” *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 4, Issue 7. 2014. P.236-239.
34. Howard S. L., Schlegel C., Iniewski K.. Error control coding in low-power wireless sensor networks: When is ECC energy-efficient?. *EURASIP Journal on Wireless Communications and Networking*, 2006. P. 29-43.
35. IEEE standard for information technology –telecommunications and information exchange between systems –local and metropolitan area networks – specific requirement part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANS). *IEEE Std. 802.15.4a-2007*, 2007. P. 203
36. Intanagonwiwat C., Govindan R., Estrin D.. Directed diffusion: a scalable and robust communication paradigm for sensor networks, *Proceedings of the ACM MobiCom'00*, Boston, MA, 2000. P. 56–67.
37. Jaikaeo C., Srisathapornphat C., Shen C.. Diagnosis of sensor networks, *IEEE International Conference on Communications ICC'01*, Helsinki, Finland, June 2001.

38. Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal, "Wireless sensor network survey", *computer networks*, ISSN: 1389-1286, Vol. 58, April 2008. P. 2292-2330.
39. Kahn J.M., Katz R.H., Pister K.S.J.. Next century challenges: mobile networking for smart dust, *Proceedings of the ACM MobiCom'99*, Washington, USA, 1999, P. 271–278.
40. Kasianchuk M . Conception of theoretical bases of the accomplished form of Krestenson's transformation and its practical application. *Proceedings of the 4-th International Conference "Advanced Computer Systems and Networks: Design and Application" (ACSN–2009)*, (Lviv, 2009). Lviv. 2009. P. 299–301.
41. Klues K, Liang CJM, Paek J, Musaloiu R, Levis P, Terzis A, Govindan R. TOSThread: Thread-Safe and Non-Invasive Preemption in TinyOS. *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*; Berkeley, CA, USA. 4–6 November 2009; P. 127–140.
42. Krishna H., Lin K.Y., Sun J.D. A coding theory approach to error control in redundant residue number systems. Part I: Theory and single error correction. *IEEE Trans. Circuits Syst.* 39, 1992. P. 8–17
43. Levis P, Madden S, Polastre J, Szewczyk R, Whitehouse K, Woo A, Gay D, Hill J, Welsh M, Brewer E, Culler D. Tinyos: An Operating System for Sensor Networks. URL: http://dx.doi.org/10.1007/3-540-27139-2_7 (17 April, 2011)
44. Lou W., Liu W., Fang Y. SPREAD: Enhancing data confidentiality in mobile ad hoc networks, *IEEE INFOCOM 2004*, HongKong, China, March, 2004.
45. Mahmood M.A., Seah W.K., & Welch I. (2015). Reliability in wireless sensor networks: A survey and challenges ahead. *Computer Networks*, 79. P. 166-187.
46. Molahosseini A. S., De Sousa, L. S., Chang, C. H. *Embedded systems design with special arithmetic and number systems*. Springer. 2017. P. 390

47. Neelam Srivastava. "Challenges of Next-Generation Wireless Sensor Networks and its Impact on Society", *Journal of Telecommunications*, Vol. 1, Issue 1, 2010. P.128-133.
48. Noury N., Herve T., Rialle V., Virone G., Mercier E., Morey G., Moro A., Porcheron T.. Monitoring behavior in home using a smart fall sensor, *IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology*, October 2000, P. 607–610.
49. Nykolaychuk Ya.M., Kasianchuk M.M., Yakymenko I.Z.. Theoretical foundations for the analytical computation of coefficients of basic numbers of Krestenson's transformation, *Cybernetics and Systems Analysis*, (50) (2014). P. 649-654.
50. Omondi A. *Residue Number System: Theory and Implementation* / A.Omond, B.Premkumar. Imperial College Press, 2007. Vol. 2. P. 296
51. Park H., Lee J., Oh S., Yim Y., Kim S. and Nam K., "QERP: Qualitybased event reliability protocol in wireless sensor networks," in *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, Nevada, USA, 09-12 January 2011. P. 730–734.
52. Pawar, M., & Agarwal, J. A literature survey on security issues of WSN and different types of attacks in network." *Indian Journal of Computer Science and Engineering*, 2017. P. 80-83.
53. Petriu E.M., Georganas N.D., Petriu D.C., Makrakis D., Groza V.Z.. Sensor-based information appliances, *IEEE Instrumentation and Measurement Magazine* (December 2000). P.31–35.
54. Proakis J.G., *Digital Communications*, 3rd Ed. New York: McGraw-Hill, 1995.
55. Pundir S., Bhalla A., Bakshi A., Singh A.K., & Singh,D.P.. MODTEEN: Modified thresholdsensitive energy efficient network in Wireless Sensor Network. In *Advances in Computing, Communication & Automation (ICACCA)(Fall), 2017 3rd International Conference on*. IEEE. (2017, September). P. 1-5

56. Rabaey J.M., Ammer M.J., da Silva Jr. J.L., Patel D., Roundy S.. PicoRadio supports ad hoc ultra-low power wireless networking, *IEEE Computer Magazine*. 2000. P.42–48.
57. Radio modulo nRF24L01. URL: <http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01> (10 April, 2017).
58. Raspberry Pi. URL: https://en.wikipedia.org/wiki/Raspberry_Pi (20 Murch, 2017).
59. Sachenko A., Yatskiv V., Tsavolyk T. Modeling the Wireless Sensor Networks Using the Error Control Scheme. Proceedings of the 3 rd IDAACS Symposium Wireless Systems within the IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS-SWS (Offenburg, Germany, September 26–27, 2016), Offenburg. 2016. P.122-126.
60. Sieck, V. Yatskiv, A. Sachenko, T. Tsavolyk. Two-Dimensional Error Control Based on Modular Corrective Codes. *International Journal of Computing*, 2015. Vol. 14, Issue 4. P. 208-213.
61. Steele R. and Hanzo L., *Mobile Radio Communications*, 2nd Ed. IEEE Press, New York, 1999.
62. Su Jun, Yatskiv V.. Method and Device for Image Coding & Transferring Based on Residue Number System // *Sensors & Transducers Journal*, 2013. Vol.18, Special Issue. P.60-65.
63. Sun J-D., Krishna H. “A coding theory approach to error control in redundant residue number systems - Part II: multiple error detection and correction,” *IEEE Trans. Circuits Syst.*, Vol. 39, No. 1, Jan. 1992. P. 18-34.
64. Szabó N.S., Tanaka R.I. *Residue arithmetic and its applications to computer technology*, New York:McGraw-Hill, 1967. P. 27–32.
65. Tay T.F., Chang C.H. A new algorithm for single residue digit error correction in Redundant Residue Number System, in *2014 Int. Symp. on Circuits and Syst.*, Melbourne, Australia, 2014. P. 1748–1751

66. Tay T.F., Chang C.H. A non-iterative multiple residue digit error detection and correction algorithm in RRNS. *IEEE Trans. Compt.* **65**(2), 2015. P.396–408
67. Warneke B., Liebowitz B., Pister K.S.J.. Smart dust: communicating with a cubic-millimeter computer, *IEEE Computer* (January 2001). P. 2–9.
68. Watson, R. W., & Hastings, C. W. Self-checked computation using residue arithmetic. *Proceedings of the IEEE*. 1966. *54* (12), P. 1920-1931.
69. Wireless sensor network. URL: https://en.wikipedia.org/wiki/Wireless_sensor_network (30 August, 2016).
70. Yang L.L. and Hanzo L.. “RNS arithmetic assisted M modulation,” *IEEE Commun. Letters*, Vol. 3, No.2, Feb. 1999. P. 28-30.
71. Yatskiv V. Tsavolyk T., Hu Zhengbing. Multiple Error Detection and Correction Based on Modular Arithmetic Correcting Codes. *Proceedings of the 8-th 2015 IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS’2015*, (Warsaw, Poland, September 24-26, 2015). Warsaw. 2015. Volume 2. P. 850-854
72. Yatskiv V. Tsavolyk T., Sachenko A. Error Correction Technique Based on Modular Correcting Codes. *Conference Proceedings IEEE 36th International Conference on Electronics and Nanotechnology (ELNANO)*, (Kyiv, Ukraine, April 19-21, 2016). Kyiv. 2016. P.362-364.
73. Yatskiv V., Tsavolyk T. Improvement of Data Transmission Reliability in Wireless Sensor Networks on the Basis of Residue Number System Correcting Codes Using the Special Module System. *Conference Proceedings, IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON) Igor Sikorsky Kyiv Polytechnic Institute*, (Kyiv, Ukraine, May 29 – June 2, 2017). Kyiv. 2017. P. 890 – 893.
74. Yatskiv V., Tsavolyk T. Two-Dimensional Corrective Codes Based on Modular Arithmetic. *Proceedings of the XIIIth International Conference the Experience of Designing and Application of CAD System in Microelectronics*

CADSM' 2015. (Lviv-Polyana, Ukraine, February 24-27, 2015). Lviv-Polyana. 2015. P. 291-294.

75. Yatskiv V., Tsavolyk T., Yatskiv N. Burst error-correcting codes based on modular correcting codes. Conference Proceedings of 14 th International Conference Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET-2018) (Lviv-Slavske, Ukraine February 20-24 2018). Lviv. 2018. P. 389-392.

76. Yatskiv V., Tsavolyk T., Yatskiv N. The Correcting Codes Formation Method Based on the Residue Number System. Conference Proceedings of 14 th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-2017) (Polyana-Svalyava, Ukraine, February 21-25, 2017). Polyana-Svalyava. 2017. P. 237-240.

77. Yatskiv V., Yatskiv N., Sachenko A., Su Jun. CPLD Encoder and Decoder for Modified Correction Codes Based on Residue Number System // Modern Problems of Radio Engineering, Telecommunications, and Computer Science. Proceedings of the International Conference TCSET'2014 Lviv-Slavske, Ukraine February 25 March 1, 2014. P. 492-493.

78. Yatskiv V., Yatskiv N., Su J., Sachenko A., Hu Z.. The use of modified correction code based on residue number system in WSN, in IEEE Int. Conf. Intelligent Data Acquisition and Advanced Computing Syst. (IDAACS) 2013, Berlin, Germany, 2013, P. 513–516.

79. Yatskiv V., Yatskiv N., Su Jun, Sachenko A., Zhengbing Hu. The Use of Modified Correction Code Based on Residue Number System in WSN // Proceedings of the 7-th 2013 IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS'2013, Berlin, Germany, 2013, Volume 1. P. 513-516.

80. Yatskiv V., Yatskiv N., Su Jun, Sachenko A., Zhengbing Hu.. The Use of Modified Correction Code Based on Residue Number System in WSN // Proceedings of the 7-th 2013 IEEE International Conference on Intelligent Data

Acquisition and Advanced Computing Systems, IDAACS'2013, Berlin, Germany, 2013, Volume 1. P. 513-516.

81. Yetgin H., Cheung K.T.K., El-Hajjar M., & Hanzo L.H.. A survey of network lifetime maximization techniques in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 2017, P. 828-854.

82. Yick J, Mukherjee B, Ghosal D. Wireless sensor network survey. *Comput Netw* 2008. 52(12). P. 2292–2330.

83. Zhang H., Arora A., Choi Y-ri. and Gouda M.G., “RBC: Reliable bursty convergecast in wireless sensor networks,” *Computer Communications*, vol. 30, no. 13, 2007. 46. P. 2560-2576

84. Zhang X., Adaptive control and reconfiguration of mobile wireless sensor networks for dynamic multi-target tracking. *IEEE Trans Autom Control*, 2011. No56(10). P. 2429–2444.

85. Zhengbing Hu, Yatskiv V., Sachenko A.. Increasing the Data Transmission Robustness in WSN Using the Modified Error Correction Codes on Residue Number System. *Elektronika ir Elektrotechnika*. Vol 21, No 1, 2015. P. 76-81.

86. Акушский И. Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. М.: Сов. радио. 1968. С. 460

87. Акушский И.Я., Пак И.Т. Вопросы помехоустойчивого кодирования в непозиционном коде // Вопросы кибернетики. 1977, Т.28. С. 36-56.

88. Алгулиев Р.М., Фаталиев Т.Х., Агаев Б.С., Алиев Т.С. Сенсорные сети: состояние, решения и перспективы. // Телекоммуникации. Ежемесячный научно-технический информационно-аналитический и учебно-методический журнал. 2007. No4. С. 27–32.

89. Беспроводные линии связи и сети.: Пер. с англ. М.: Издательский дом «Вильямс», 2003. С.640.

90. Волинський О.І. Методи високопродуктивних перетворень великорозрядних чисел з базису Радемахера у базис Крестенсона. Вісник

національного університету Львівська політехніка. Комп'ютерні системи та мережі. Львів. 2012. С. 39-48.

91. Волинський О.І. Методи порівняння та сумування в розмежованій системі числення // Поступ в науку. Збірник наукових праць Бучацького інституту менеджменту і аудиту. Бучач. 2009. №5. Т1. С. 91-94.

92. Гераїмчук М.Д., Івахів О.В., Паламар М.І., Шевчук Б.М. Основи побудови перспективних безпроводових сенсорних мереж. Монографія. К.: ЕКМО, 2010. С. 124

93. Касянчук М.М., Якименко І.З., Паздрій І.Р., Николайчук Я.М.. Аналітичний пошук модулів досконалої форми системи залишкових класів та їх застосування в китайській теоремі про залишки // Вісник Хмельницького національного університету. Технічні науки. 2015. № 1 (221). С. 170 – 176.

94. Касянчук М.М.. Побудова модифікованої досконалої форми системи залишкових класів на основі розв'язку систем конгруенцій // Науковий вісник НЛТУ України. 2016. Вип. 26 (7). С. 372 – 377.

95. Касянчук М.Н.. Концепція теоретичних положень досконалої форми перетворення Крестенсона і його практичне застосування // Оптико-електронні інформаційно-енергетичні технології. 2010. № 2 (20). С. 43 – 48.

96. Краснобаєв В.А., Кошман С.О., Мавріна М.О.. Метод підвищення достовірності контролю даних, представлених у системі залишкових класів // Кибернетика и системний аналіз. 2014. Том 50, №6. С.167 –175.

97. Мерекин Ю. В. Нижняя оценка сложности для схем конкатенации слов // Дискретный анализ и исследование операций. 1996. Том 3, № 1. С.52-56.

98. Минович А.И., Романюк В.А. Маршрутизация в мобильных радиосетях –проблема и пути решения // Зв'язок. 2006. № 7. С. 49 –55.

99. Міночкін А.І., Романюк В.А., Жук О.В. Перспективи розвитку сенсорних мереж // Зв'язок. 2008. № 1. С. 16 – 21.

100. Николайчук Я.М. Теорія джерел інформації. Тернопіль: ТЗОВ „Терно–граф”. 2010. С. 536.

101. Николайчук Я.Н., Касянчук М.Н., Якименко И.З.. Теоретические основы модифицированной совершенной формы системы остаточных классов // Кибернетика и системный анализ. 2016. Т. 52, № 2. С. 51-55.

102. Патент України на корисну модель № 117493, МПК (2006) H04J 13/00 H04W 84/18 (2009.01). Спосіб формування корегуючих кодів в системі залишкових класів / В. В. Яцків, Т. Г. Цаволик, Н. Г. Яцків. – № u201700839; заявл. 30.01.2017; опубл. 26.06.2017. – Бюл. № 12/2017.

103. Питерсон У., Уэлдон Э. Коды исправляющие ошибки.: Мир, 1976. С. 593

104. Скрипов С.А. Разработка протоколов маршрутизации для беспроводных сетей со специальной топологией // IV Международная научно-практическая конференция "Современные информационные технологии и ИТ бразования". 2009.

105. Цаволик Т. Г. Корегуючі коди в системі залишкових класів зі спеціальними модулями // Вимірювальна та обчислювальна техніка в технологічних процесах. Технічні науки. 2016. № 3. С. 100 – 104.

106. Цаволик Т. Г. Коректуючі коди системи залишкових класів зі спеціальною системою модулів. Матеріали V Всеукраїнської школи-семінару молодих вчених і студентів «Сучасні комп'ютерні інформаційні технології» (м. Тернопіль, 22-23 травня 2015 р.). Тернопіль: ТНЕУ. 2015. С. 61-62.

107. Цаволик Т. Г. Яцків В. В. виправлення пакетів помилок на основі модулярного коригуючого коду // Науковий вісник НЛТУ України. 2018. Вип. 28(2). С. 155–158.

108. Цаволик Т. Г., Яцків В. В. Метод формування корегувальних кодів у системі залишкових класів // Науковий вісник НЛТУ України. 2017. Вип. 27 (3). С. 191–194.

109. Цаволик Т. Г., Яцків В. В. Метод исправления ошибок на основе модулярных корректирующих кодов // Физика, математика, информатика.

Вестник Брестского государственного технического университета. Брест, 2015. № 5 (850). С. 36-38.

110. Яцків В. В. Теоретичні основи створення і структурна організація компонентів безпроводних сенсорних мереж підвищеної ефективності: дис. ... д-ра техн. наук. Тернопіль, 2016. С. 327

111. Яцків В. В. Виявлення та виправлення багатократних помилок на основі модулярних коректуючих кодів // Інформаційні технології та комп'ютерна інженерія. 2015. Том 33, № 2. С. 77–82.

112. Яцків В. В., Цаволик Т. Г. Метод завадостійкого кодування даних на основі модулярних коректуючих кодів. Матеріали V Міжнародної науково-практичної конференції молодих вчених «Інфокомунікації – сучасність та майбутнє» (м. Одеса, 29-30 жовтня 2015 р.). Одеса, ОНАЗ. 2015. С. 111-114.

113. Яцків В. В., Цаволик Т. Г. Двовимірні коректуючі коди на основі модулярної арифметики // Вісник Хмельницького національного університету. Технічні науки. 2015. № 4 (227). С. 144 – 148.

114. Яцків В. В., Цаволик Т. Г., Стасюк І. І. Адаптивний метод контролю помилок в безпроводних сенсорних мережах. Прикладні науково-технічні дослідження: матеріали Міжнародної науково-практичної конференції (м. Івано-Франківськ, 5-7 квітня 2017 р.) Івано-Франківськ: Симфонія форте. 2017. С. 30.

115. Яцків В.В. Виявлення помилок на основі модифікованих коректуючих кодів системи залишкових класів // Тези доповідей Четвертої науково-практичної конференції «Методи та засоби кодування, захисту й ущільнення інформації». Вінниця: ПП «ТД Едельвейс і К», 2013. С. 40-43.

116. Яцків В.В. Модифіковані коректуючі коди системи залишкових класів та їх застосування // Інформаційні технології та комп'ютерна інженерія. 2013. №2. С.39-45.

Додаток А

Лістинг програми для знаходження взаємно простих коефіцієнтів

```

using System;
using System.Collections.Generic;
using System.Linq;
namespace okm
{
    class Program
    {
        static List<double[]> allRez = new
List<double[]>();

        public static void Show(double P) {
            for (int i = 0; i < allRez.Count; i++)
            {
                for (int j = 1; j < allRez.Count;
j++)
                {
                    if (allRez[i] == allRez[j])
                    {
                        allRez.Remove(allRez[j]);
                    }
                }
            }
            foreach (double[] el in allRez)
            {
                for (int i = 0; i < el.Length; i++)
                {
                    Console.Write(el[i] + " ");
                }
                Console.WriteLine("P is equal to:
" + P);
            }
        }

        public static void District() {
            foreach (double[] a in allRez) {
                Array.Sort(a);
            }
            List<double[]> uniq = new
List<double[]>(allRez.Distinct());
            allRez = uniq;
        }

        public static void Introduce(double[] V,
int[] mass)
        {
            double[] arr = new
double[mass.Length];
            for (int i = 0; i < mass.Length; i++)
            {
                arr[i]=V[mass[i]];
            }
            allRez.Add(arr);
        } //результати записуються в список.
        public static int[] makeZ(int[] a) {
            for(int i=0;i<a.Length;i++) {
                a[i] = 0;
            }
            return a;
        }
        public static double mod(double a,
double b)
        {
            if ((a < 0)&& a%b!=0)
            {
                var q =
Math.Abs(Convert.ToInt32(a/b))+1;
                var c = (b*(q + 1) + a);
                var d=Convert.ToInt32(c/b);
                c = c>b*d?(c-b*d):c-b*(d-1);
                return c;
            }
            else
            {
                return a%b;
            }
        }
        public static List<double>
GenerateSimpleNumbers(double P)
        {
            var list = new List<double>();
            bool yes;
            for (int n = 2; n <= P; n++)
            {

```

```

yes = true;
for (int m = 2; m <= n / 2; m++)
    if (n % m == 0)
        {
            yes = false;
            break;
        }
    if (yes) list.Add(n);
}
return list;
}
public static double SimpleP(double a,
double[] b)
{
    for (int i = 0; i < b.Length; i++)
        {
            if (b[i] >= a)
                {
                    a = b[i];
                    break;
                }
        }
    return a;
}
} //для знаходження мінімального
простого P
public static bool
AreEqual(List<double> a)
{
    List<double> uniq = new
List<double>(a.Distinct());
    if (uniq.Count == a.Count)
        {
            return false;
        }
    else
        return true;
}
} //Перевірка на Повторюваність
елементів
public static double[]
GenerateN(double n)
{
    double[] arr = new
double[Convert.ToInt32(n)];
    for (int i = 0; i < arr.Length / 2; i++)
        {
            arr[i] = i + 1;
            arr[Convert.ToInt32(n - 1 - i)] = -
(i + 1);
        }
    return arr;
}
} //Створення масиву значень X
public static double[] action(double N,
double[] M, double P)
{
    double[] arr = new
double[M.Length];
    for (int i = 0; i < arr.Length; i++)
        {
            arr[i] = mod((N * M[i]), P);
        }
    return arr;
}
} //виконання матем. операції
public static int[] Increaser(int[] mass,
int SimpleNumb)
{
    for (int i = mass.Length - 1; i >= 0; i-
-)
        {
            mass[i]++;
            if (mass[0] >= SimpleNumb)
                {
                    Console.WriteLine("no more
elements to show");
                    // Console.ReadKey();
                    // Environment.Exit(0);
                    mass[0] = SimpleNumb+1;
                    return mass;
                }
            if (mass[i] >= SimpleNumb) {
mass[i] = mass[0] + 1; }
            else { break; }
        }
    return mass;
}
} //функція перебирає всі можливі
комбінації елементів масиву
static void Main(string[] args)
{
    int counter = 0,m,n;

```

```

        Console.WriteLine("Введіть
кількість розрядів: ");
        var
firstcheck=int.TryParse(Console.ReadLine()
,out n);//ввід розрядності
        Console.WriteLine("Введіть
кількість масивів: ");
        var secondcheck=
int.TryParse(Console.ReadLine(),out
m);//ввід кількості масивів
        double P=0, sqr = Math.Pow(2, n +
1) - 2;
        double[] X = GenerateN(sqr);//масив
розбросу чисел залежно івд розрядності
        int[] mass = new int[m];
        while (counter == 0)
        {
            Console.WriteLine("Введіть P: ");
            var
thirdcheck=double.TryParse(Console.ReadL
ine(),out P);
            if (!firstcheck || !secondcheck ||
!thirdcheck)
            {
                Console.WriteLine("Error!
There are empty field, try again next time!");
                Console.ReadKey();
                Environment.Exit(0);
            }
            List<double> V
=GenerateSimpleNumbers(P);//масив
дійсних
            int SimpleNumb = V.Count;
            mass = makeZ(mass);

```

```

        while (true)
        {
            List<double> list = new
List<double>();
            for (int i = 0; i < mass.Length;
i++)
            {
                list.AddRange(action(V[mass[i]], X, P));
            }
            if (AreEqual(list))
            {
                Increaser(mass,
SimpleNumb);
            }
            else
            {
                counter++;
                Introduce(V.ToArray(),
mass);
                mass = Increaser(mass,
SimpleNumb);
            }
            if (mass[0] ==
SimpleNumb+1) { break; }
        }
        }
        District();
        Show(P);
        Console.ReadKey();
    }
}

```

Додаток Б

Декодер виявлення та виправлення багатократних помилок

```

mass=[
  [10, 4, 3, 4, 8, 5, 4, 10],
  [9, 8, 4, 9, 7, 3, 8, 8],
  [9, 1, 3, 8, 7, 3, 6, 10],
  [4, 4, 10, 7, 2, 9, 10, 6],
  [5, 2, 9, 2, 3, 1, 4, 4],
  [3, 9, 9, 8, 8, 9, 2, 8],
  [2, 6, 6, 2, 7, 4, 6, 6],
  [8, 4, 4, 1, 10, 5, 6, 7]
]

size=8
P=131297
v=[211,257,263,269,271,277,283,288]
#=====
mass2=[
  [10, 4, 3, 4, 8, 5, 4, 10],
  [9, 8, 4, 9, 7, 3, 8, 8],
  [9, 1, 3, 8, 7, 3, 6, 10],
  [4, 4, 10, 7, 2, 9, 10, 6],
  [5, 2, 9, 2, 3, 1, 4, 4],
  [3, 9, 9, 8, 8, 9, 2, 8],
  [2, 6, 6, 2, 7, 4, 6, 6],
  [8, 10, 4, 1, 10, 5, 6, 7]
]
v=[211,257,263,269,271,277,283,288]
#===== ----- функція update
def update(mass):
  rez = 0
  for i in range(size):
    for j in range(size):
      rez += mass[i][j]*v[i]
    rez = rez % P
    mass[i].append(rez)
    rez = 0
  o_mass = []
  for i in range(size):
    for j in range(size):
      rez += mass[j][i]*v[j]
    rez = rez % P

```

```

    o_mass.append(rez)
    rez = 0
    mass.append(o_mass)
    print("Mass = ")
    for i in range(size + 1):
        print(mass[i])
#=====
update(mass)
print("=====")
update(mass2)
#=====
def search_mistake(mass,mass2):
    Error_first_index = -1
    Error_last_index = -1
    for i in range(size):
        if (mass[i][size]-mass2[i][size]!=0):
            Error_first_index = i
        if (mass[size][i]-mass2[size][i]!=0):
            Error_last_index = i
    Error = [Error_first_index,Error_last_index]
    return Error
tmp = search_mistake(mass,mass2)
first = int(tmp[0])
last = int(tmp[1])
print(first,last)
#=====
def decision():
    mass2[first][last] = (v[first]*(mass[first][last]-mass2[first][last]))%P
decision()
print("output mass===")
for i in range(size + 1):
    print(mass2[i])

```

Додаток В

Акти про використання результатів дисертаційного дослідження

ТОВ «Інтелдім»
вул. Р.Барвінських, 6
46001, м. Тернопіль, Україна

intelDim
Інжинірингова компанія

LLC «Inteldim»
Rodyny Barvinskyh st. 6
46001, Ternopil, Ukraine

тел./факс +380 (352) 22-05-55
тел. +380 (67) 353-31-63

info@inteldim.com.ua
www.inteldim.com.ua

tel./fax +380 (352) 22-05-55
tel. +380 (67) 353-31-63

ЗАТВЕРДЖУЮ
Директор ТОВ «Інтелдім»
_____ Скалецький О. В.
« 12 » _____ 02 2018 р.

АКТ


про впровадження результатів дисертаційної роботи
Цаволика Тараса Григоровича
на тему «**Методи та засоби передачі даних в безпроводних сенсорних
мережах на основі модулярних коригуючих кодів**»

Ми, комісія в складі: А.В.Фік та І.В.Скалецький, склали даний акт про те, що результати дисертаційної роботи Цаволика Т.Г. на тему «Методи та засоби передачі даних в безпроводних сенсорних мережах на основі модулярних коригуючих кодів» використані при розробці безпроводної системи збору даних з розподілених сенсорів, зокрема:

- метод та алгоритм виправлення помилок в двох інформаційних символах на основі модулярних коригуючих кодів з одним перевіроочним символом;
- метод виправлення пакетів помилок на основі модулярних коригуючих кодів.

Застосування вказаних методів дозволило підвищити надійність передачі даних в безпроводній системі збору даних з розподілених сенсорів інтелектуальних будинків.

Головний інженер


І.В.Скалецький

Виконавець робіт з налагоджування
та випробовування


А.В.Фік



ТОВ «Протекшн-Груп»,
код за ЄДРПОУ 38422777
47772, Тернопільська область, Тернопільський район, с.
Великі Гаї, вул. Галицька, буд. 42Є
р/р № 26001300610262 в ТБВБ №10019/03 філії –
Тернопільське обласне управління АТ «Ощадбанк»,
МФО 338545
Індивідуальний податковий номер 384227719103

№ 34/04-1 від 23.04.2018р

на № _____

ЗАТВЕРДЖУЮ
Директор ТОВ «Протекшн-Груп»
Сірант О. Ю.
« _____ » _____ 2018 р

АКТ

про впровадження результатів дисертаційної роботи

Цаволика Тараса Григоровича на тему «Методи та засоби передачі даних в безпроводних сенсорних мережах на основі модулярних коригуючих кодів»,
представленої на здобуття наукового ступеня кандидата технічних наук

Даний акт складений у тому, що результати дисертаційної роботи Цаволика Т. Г. на тему «Методи та засоби передачі даних в безпроводних сенсорних мережах на основі модулярних коригуючих кодів» передані для впровадження компанії «Протекшн-Груп», що включають:

- метод формування перевірочних символів в коригуючих кодах системи залишкових класів;
- адаптивний протокол передачі даних в безпроводних сенсорних мережах.

Використання вказаних наукових результатів забезпечило зниження енергозатрат та підвищення надійності передачі даних в безпроводних каналах зв'язку і впроваджені при розробці систем «Безпечне місто» та «Розумний будинок».

Директор ТОВ «Протекшн-Груп»

ТОВ «Протекшн-Груп»

Сірант О. Ю.

ЗАТВЕРДЖУЮ
Проректор з наукової роботи
Тернопільського національного
економічного університету
д.е.н., проф. Задорожний З.-М. В.



2018 р.

АКТ

про використання результатів кандидатської дисертації
Цаволика Тараса Григоровича на тему:
«Методи та засоби передачі даних в безпроводних сенсорних мережах на основі
модулярних коригуючих кодів»


Комісія у складі: голови – декана факультету комп'ютерних інформаційних технологій, д.т.н., проф. Дивака М.П. та членів: в.о.директора науково-дослідного інституту інноваційного розвитку та державотворення Письменного В.І., завідувача кафедри спеціалізованих комп'ютерних систем, д.т.н., професора Николайчука Я.М., завідувача кафедри кібербезпеки, д.т.н., доцента Яцківа В.В. склали цей акт про те, що дослідження та результати дисертаційної роботи Цаволика Т.Г. використані під час виконання науково-дослідних робіт на кафедрах спеціалізованих комп'ютерних систем та кібербезпеки факультету комп'ютерних інформаційних технологій з безпосередньою участю автора, а саме:

1) науково-дослідної роботи на тему “Методи та засоби побудови безпроводних мультимедійних сенсорних мереж на основі модулярної арифметики” (номер державної реєстрації 0112U007886), у якій автором розроблено алгоритм виправлення помилок на основі модулярних коригуючих кодів та алгоритм пошуку взаємно простих коефіцієнтів для побудови модулярних коригуючих кодів;

2) науково-дослідної роботи на тему “Теоретичні основи та апаратні засоби підвищення продуктивності роботи безпроводних сенсорних мереж” (номер державної реєстрації 0117U000414), у якій автором розроблено метод формування перевірочних символів у коригуючих кодах системи залишкових класів та метод виправлення пакетів помилок на основі модулярного коригуючого коду.

Голова комісії


декан факультету комп'ютерних
інформаційних технологій,
д.т.н., професор


Дивак М.П.
Члени комісії:

в.о. директора НДІ ІРД


Письменний В.І.

завідувач кафедри спеціалізованих
комп'ютерних систем,
д.т.н., професор


Николайчук Я.М.

завідувач кафедри кібербезпеки,
д.т.н., доцент


Яцків В.В.

ЗАТВЕРДЖУЮ

Перший проректор
Тернопільського національного
економічного університету

Шинкарик М.І.

« 02 2018 р.

АКТ

про впровадження результатів дисертаційної роботи
Цаволика Тараса Григоровича**“Методи та засоби передачі даних в безпроводних сенсорних мережах на основі модулярних коригуючих кодів”**

Комісія у складі декана факультету комп'ютерних інформаційних технологій Тернопільського національного економічного університету (ТНЕУ), д.т.н., проф. Дивака Миколи Петровича (голова комісії), завідувача кафедри комп'ютерної інженерії ТНЕУ, д.т.н., проф. Березького Олега Миколайовича (член комісії), завідувача кафедри кібербезпеки ТНЕУ, д.т.н., доцента Яцківа Василя Васильовича (член комісії), підтверджує, що результати кандидатської дисертації Цаволика Т. Г. впроваджені і використовуються в навчальному процесі при підготовці бакалаврів та магістрів на кафедрах комп'ютерної інженерії та кібербезпеки Тернопільського національного економічного університету при вивченні дисциплін:

1. Теорія інформації та кодування:
 - метод формування коригуючих кодів в системі залишкових класів;
 - методи виявлення та виправлення помилок на основі модулярних коригуючих кодів.
2. Проектування в середовищі Інтернет – речей:
 - методи підвищення надійності передачі даних в середовищі Інтернет – речей;
 - адаптивний алгоритм та протокол передачі даних в безпроводних сенсорних мережах.
3. Програмування комп'ютерних систем на програмованих логічних інтегральних схемах:
 - моделі кодерів / декодерів завадостійкого кодування на мові Verilog;
 - реалізація кодерів / декодерів завадостійкого кодування на програмованих логічних інтегральних схемах.

Декан факультету комп'ютерних
інформаційних технологій,
д.т.н., професор

Дивак М.П.

Завідувач кафедри
комп'ютерної інженерії, д.т.н., професор

Березький О.М.

Завідувач кафедри
кібербезпеки, д.т.н., доцент

Яцків В.В.

Додаток Г

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМЮ ДИСЕРТАЦІЇ**Наукові праці, в яких опубліковані основні наукові результати****дисертації:**

1. Цаволик Т. Г., Яцків В. В. Метод исправления ошибок на основе модулярных корректирующих кодов // Физика, математика, информатика. Вестник Брестского государственного технического университета. Брест, 2015. № 5 (850). С. 36-38.
2. Цаволик Т. Г., Яцків В. В. Метод формування корегувальних кодів у системі залишкових класів // Науковий вісник НЛТУ України. 2017. Вип. 27 (3). С. 191–194.
3. Яцків В. В., Цаволик Т. Г. Двовимірні коректуючі коди на основі модулярної арифметики // Вісник Хмельницького національного університету. Технічні науки. 2015. № 4 (227). С. 144 – 148.
4. Sieck, V. Yatskiv, A. Sachenko, T. Tsavolyk. Two-Dimensional Error Control Based on Modular Corrective Codes. International Journal of Computing, 2015. Vol. 14, Issue 4. P. 208-213.
5. Цаволик Т. Г. Корегуючі коди в системі залишкових класів зі спеціальними модулями // Вимірювальна та обчислювальна техніка в технологічних процесах. Технічні науки. 2016. № 3. С. 100 – 104.
6. Цаволик Т. Г. Яцків В. В. виправлення пакетів помилок на основі модулярного коригуючого коду // Науковий вісник НЛТУ України. 2018. Вип. 28(2). С. 155–158.
7. Патент України на корисну модель № 117493, МПК (2006) H04J 13/00 H04W84/18 (2009.01). Спосіб формування корегуючих кодів в системі залишкових класів / В. В. Яцків, Т. Г. Цаволик, Н. Г. Яцків. – № u201700839; заявл. 30.01.2017; опубл. 26.06.2017. – Бюл. № 12/2017.

Опубліковані праці апробаційного характеру:

1. Yatskiv V. Tsavolyk T., Hu Zhengbing. Multiple Error Detection and Correction Based on Modular Arithmetic Correcting Codes. Proceedings of the 8-th 2015 IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS'2015, (Warsaw, Poland, September 24-26, 2015). Warsaw. 2015. Volume 2. P. 850-854.
2. Yatskiv V., Tsavolyk T. Two-Dimensional Corrective Codes Based on Modular Arithmetic. Proceedings of the XIIIth International Conference the Experience of Designing and Application of CAD System in Microelectronics CADSM' 2015. (Lviv-Polyana, Ukraine, February 24-27, 2015). Lviv-Polyana. 2015. P. 291-294.
3. Yatskiv V. Tsavolyk T., Sachenko A. Error Correction Technique Based on Modular Correcting Codes. Conference Proceedings IEEE 36th International

Conference on Electronics and Nanotechnology (ELNANO), (Kyiv, Ukraine, April 19-21, 2016). Kyiv. 2016. P.362-364.

4. Sachenko A., Yatskiv V., Tsavolyk T. Modeling the Wireless Sensor Networks Using the Error Control Scheme. Proceedings of the 3 rd IDAACS Symposium Wireless Systems within the IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS-SWS (Offenburg, Germany, September 26–27, 2016), Offenburg. 2016. P.122-126.

5. Yatskiv V., Tsavolyk T., Yatskiv N. The Correcting Codes Formation Method Based on the Residue Number System. Conference Proceedings of 14 th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-2017) (Polyana-Svalyava, Ukraine, February 21-25, 2017). Polyana-Svalyava. 2017. P. 237-240.

6. Yatskiv V., Tsavolyk T. Improvement of Data Transmission Reliability in Wireless Sensor Networks on the Basis of Residue Number System Correcting Codes Using the Special Module System. Conference Proceedings, IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON) Igor Sikorsky Kyiv Polytechnic Institute, (Kyiv, Ukraine, May 29 – June 2, 2017). Kyiv. 2017. P. 890 – 893.

7. Yatskiv V., Tsavolyk T., Yatskiv N. Burst error-correcting codes based on modular correcting codes. Conference Proceedings of 14 th International Conference Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET-2018) (Lviv-Slavske, Ukraine February 20-24 2018). Lviv. 2018. P. 389-392.

8. Цаволик Т. Г. Коректуючі коди системи залишкових класів зі спеціальною системою модулів. Матеріали V Всеукраїнської школи-семінару молодих вчених і студентів «Сучасні комп'ютерні інформаційні технології» (м. Тернопіль, 22-23 травня 2015 р.). Тернопіль: ТНЕУ. 2015. С. 61-62.

9. Яцків В. В., Цаволик Т. Г. Метод завадостійкого кодування даних на основі модулярних коректуючих кодів. Матеріали V Міжнародної науково-практичної конференції молодих вчених «Інфокомунікації – сучасність та майбутнє» (м. Одеса, 29-30 жовтня 2015 р.). Одеса, ОНАЗ. 2015. С. 111-114.

10. Яцків В. В., Цаволик Т. Г., Стасюк І. І. Адаптивний метод контролю помилок в безпроводних сенсорних мережах. Прикладні науково-технічні дослідження: матеріали Міжнародної науково-практичної конференції (м. Івано-Франківськ, 5-7 квітня 2017 р.) Івано-Франківськ: Симфонія форте. 2017. С. 30.

Опубліковані праці апробаційного характеру

№ з/п	Назва конференції	Місце проведення	Дата проведення	Форма участі
1	Proceedings of the 8-th 2015 IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS'2015	Warsaw, Poland	September 24-26, 2015	Очна
2	Proceedings of the XIII th International Conference the Experience of Designing and Application of CAD System in Microelectronics CADSM' 2015	Lviv-Polyana, Ukraine	February 24-27, 2015	Очна
3	Conference Proceedings IEEE 36th International Conference on Electronics and Nanotechnology (ELNANO)	Kyiv, Ukraine	April 19-21, 2016	Очна
4	Proceedings of the 3 rd IDAACS Symposium Wireless Systems within the IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS-SWS	Offenburg, Germany	September 26-27, 2016	Очна
5	Conference Proceedings of 14 th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-2017)	Polyana-Svalyava, Ukraine	February 21-25, 2017	Очна
6	Conference Proceedings, IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)	Kyiv, Ukraine	May 29 – June 2, 2017	Очна
7	Conference Proceedings of 14 th International Conference Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET-2018)	Lviv-Slavske, Ukraine	February 20-24 2018	Очна
8	Матеріали V Всеукраїнської школи-семінару молодих вчених і студентів «Сучасні комп'ютерні інформаційні технології»	м. Тернопіль	22-23 травня 2015 р.	Очна
9	Матеріали V Міжнародної науково-практичної конференції молодих вчених «Інфокомунікації – сучасність та майбутнє»	м. Одеса	29-30 жовтня 2015 р.	заочна
10	Прикладні науково-технічні дослідження: матеріали Міжнародної науково-практичної конференції	м. Івано-Франківськ	5-7 квітня 2017 р.	заочна