

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

На правах рукопису

ЯЦКІВ ВАСИЛЬ ВАСИЛЬОВИЧ

УДК 004.75

**ТЕОРЕТИЧНІ ОСНОВИ СТВОРЕННЯ І СТРУКТУРНА ОРГАНІЗАЦІЯ
КОМПОНЕНТІВ БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖ
ПІДВИЩЕНОЇ ЕФЕКТИВНОСТІ**

05.13.05 – комп'ютерні системи та компоненти

Дисертація на здобуття наукового ступеня
доктора технічних наук

Наукові консультанти
Саченко Анатолій Олексійович,
д.т.н., професор;
Юрген Зік,
доктор наук, професор,
Університет прикладних наук,
м.Берлін, Німеччина

Ідентичність всіх примірників дисертації

ЗАСВІДЧУЮ

Учений секретар спеціалізованої

Вченої ради

/ Я. Луцик /

Тернопіль – 2016

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП	7
РОЗДІЛ 1. СУЧАСНИЙ СТАН ТА ШЛЯХИ УДОСКОНАЛЕННЯ БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖ	17
1.1 Стан розробки безпроводних сенсорних мереж та їх застосування	17
1.2 Переваги використання програмованих логічних інтегральних схем для побудови безпроводних мультимедійних сенсорних вузлів	33
1.3 Аналіз методів підвищення корисної пропускної здатності безпроводних сенсорних мереж	39
1.4 Аналіз підходів підвищення надійності передачі даних у безпроводних сенсорних мережах	46
1.5 Аналіз критеріїв оцінки ефективності безпроводних сенсорних мереж	61
1.6 Напрями підвищення ефективності роботи безпроводних сенсорних мереж	66
Висновки до першого розділу	67
РОЗДІЛ 2 МЕТОДИ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖ НА ОСНОВІ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ	69
2.1 Метод мережного кодування даних на основі системи залишкових класів	69
2.2 Кодове розділення каналів зв'язку на основі перетворення системи залишкових класів	81
2.3 Модифікований метод стрибкоподібної зміни частоти	86
2.4 Перетворення та обробка зображень в безпроводних сенсорних мережах	92

	3
Висновки до другого розділу	99
РОЗДІЛ 3 ТЕОРЕТИЧНІ ОСНОВИ ПІДВИЩЕННЯ НАДІЙНОСТІ ПЕРЕДАЧІ ДАНИХ У БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖАХ	101
3.1 Коректуючі коди на основі модулярної арифметики	101
3.2 виправлення багатократних помилок на основі модулярних коректуючих кодів	107
3.3 Двовимірний контроль помилок на основі модулярних коректуючих кодів	122
3.4 виявлення та виправлення помилок при виконанні арифметичних операцій	129
3.5 Підвищення надійності передачі даних в безпроводних сенсорних мережах на основі декомпозиції	136
Висновки до третього розділу	141
РОЗДІЛ 4 КОНЦЕПЦІЯ ПОБУДОВИ ТА АЛГОРИТМИ ПЕРЕДАЧІ ДАНИХ У БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖАХ	143
4.1 Побудова безпроводних сенсорних мереж на основі колективного інтелекту	143
4.2 Протокол передачі даних на основі мережного кодування	150
4.3. Алгоритм адаптивної передачі даних у безпроводних сенсорних мережах	161
Висновки до четвертого розділу	163
РОЗДІЛ 5. СТРУКТУРНА ОРГАНІЗАЦІЯ ТА СИНТЕЗ КОМПОНЕНТІВ СЕНСОРНИХ МЕРЕЖ НА ПРОГРАМОВАНИХ ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМАХ	165
5.1 Експериментальні дослідження апаратної та часової складності методів виконання модулярних операцій	166
5.2 Пристрій виявлення та виправлення однократних помилок на основі модулярних коректуючих кодів	178
5.3 Синтез пристрою виправлення помилок у двох символах на основі модулярних коректуючих кодів	188

5.4 Пристрій багаторівневого перетворення даних з системи залишкових класів у позиційний код	196
5.5 Програмний модуль генератора Verilog-коду	204
Висновки до п'ятого розділу	211
РОЗДІЛ 6. РОЗРОБКА, МОДЕЛЮВАННЯ ТА ВПРОВАДЖЕННЯ УДОСКОНАЛЕНИХ БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖ	214
6.1 Моделювання роботи мережі при використанні схем контролю помилок	214
6.2 Експериментальне тестування каналу зв'язку безпроводної сенсорної мережі	220
6.3 Безпроводна сенсорна мережа системи відеоспостереження об'єктів	227
6.4 Підвищення надійності передачі даних у системі дистанційного керування технологічними процесами „СТРІЛА-М”	238
Висновки до шостого розділу	248
ВИСНОВКИ	249
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	254
ДОДАТОК А Характеристики комерційних модулів безпроводних сенсорних мереж	281
ДОДАТОК Б Текст програми пошуку коефіцієнтів	283
ДОДАТОК В Таблиця синдромів для виправлення помилок	285
ДОДАТОК Г Функціональна схема декодера	287
ДОДАТОК Д Verilog-код декодера модулярного коректуючого коду	288
ДОДАТОК Е Програмний код генератора Verilog-коду	313
ДОДАТОК Ж Акти впровадження результатів роботи	322

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БСМ – Безпроводна сенсорна мережа.

БМСМ – Безпроводна мультимедійна сенсорна мережа.

СЗК – система залишкових класів.

БЧХ – Боуз - Чоудхурі – Хоквінгом.

QoS – Quality of Service – якість обслуговування.

ARQ – Automatic Repeat reQuest – автоматичний запит повторної передачі.

HARQ – Hybrid automatic repeat request – гібридна схема повторної передачі пакетів.

LDPC – Low-density parity-check code – код з малою щільністю перевірок на парність.

FEC – Forward Error Correction – пряме виправлення помилок.

BER – Bit Error Rate – коефіцієнт бітових помилок.

PER – Packet Error Rate – кількість втрачених пакетів.

FDMA – Frequency Division Multiple Access – множинний доступ з поділом каналів за частотою.

TDMA – Time division multiple access – часове розділення каналів.

TRNS – Transformation Residue Number System – розділення каналів зв'язку на основі перетворення системи залишкових класів.

CDMA – Code Division Multiple Access – багаточисельний доступ з кодовим розділенням.

MAC – Medium Access Control – управління доступом до середовища.

ASIC – Application-Specific Integrated Circuit – спеціалізована інтегральна схема.

OQPSK – Offset Quadrature Phase-shift keying – квадратурна фазова маніпуляція зі зсувом.

PER – Packet Error Rate – кількість втрачених пакетів.

MAC – Medium Access Control - керування доступом до середовища.

CSMA-CA – Carrier Sense Multiple Access with Collision Avoidance -
множинний доступ з контролем несучої частоти та уникненням колізій.

FCS – Frame Check Sequence - контрольна сума кадру.

LAN – Local Area Network - локальна комп'ютерна мережа.

MFR – MAC Footer - завершальна частина MAC-кадру.

MHR – MAC Header - MAC-заголовок.

PAN – Personal Area Network - персональна мережа.

PDU – Protocol Data Unit - блок даних протоколу.

ВСТУП

Актуальність теми. Прогрес у галузі мікроелектроніки та безпроводних технологій передачі даних став основою для створення нового класу розподілених комп'ютерних систем – безпроводних сенсорних мереж (БСМ), які набувають дедалі ширшого застосування. БСМ є одним із сучасних і перспективних напрямків розвитку розподілених самоконфігурованих систем моніторингу й управління ресурсами та процесами.

Вузлами БСМ є малогабаритні пристрої, що виконують одночасно вимірювальні, обчислювальні й комунікаційні функції. Характерна їхня особливість полягає в обмеженості обчислювальних і комунікаційних ресурсів та вимог до тривалої роботи від автономного джерела живлення. Безпроводні вузли, як правило, не обмінюються повідомленнями між собою, а тільки передають дані своїх сенсорів і ретранслюють дані сенсорів інших вузлів на базову станцію. При цьому в системах моніторингу окремі вузли можуть знаходитись на значній відстані від базової станції, що висуває додаткові вимоги до протоколів маршрутизації та надійності передачі даних. Особливо гостро ця проблема стосується БСМ, які використовуються для збору й обробки мультимедійних даних.

Передача даних у БСМ – це найбільш енергозатратна операція, яка використовує понад 70% енергії, натомість локальна обробка даних у безпроводних вузлах дасть змогу зменшити енергозатрати на передачу даних за рахунок використання ефективних методів обробки та протоколів передачі даних.

Водночас використання БСМ у системах критичного застосування ставить підвищені вимоги до надійності та продуктивності функціонування на всіх рівнях взаємодії еталонної моделі відкритих систем.

З огляду на те, що БСМ використовують відкрите середовище передачі сигналів, важливою залишається проблема забезпечення високої надійності

передачі даних, вирішення якої сприятиме підвищенню ефективності функціонування БСМ загалом.

Однак у процесі створення БСМ виникає така суперечність: у загальному випадку застосування методів обробки та передачі даних, що забезпечують високу надійність передачі, призводить до зниження продуктивності мережі або зростання обчислювальної складності і, відповідно, підвищення енергозатрат.

Існуючі підходи та методи обробки даних, які функціонують у позиційних системах числення, не можуть забезпечити підвищені вимоги до надійності передачі даних без зниження продуктивності мережі при обмежених апаратних ресурсах і використанні автономного живлення.

Значний теоретичний внесок у розвиток системи залишкових класів та її застосування в обчислювальній техніці зробили такі вчені: І. Я. Акушський, Д. І. Юдицький, В. М. Амербаєв, В. А. Торгашев, А. О. Коляда, В. А. Краснобаєв, Я. М. Николайчук, М. І. Червяков, О. А. Фінько, А. Омонді (A. Omondi), Б. Премкумар (B. Premkumar), Дж. Кардарілі (G. Cardarilli). Дослідження, що сприяли розвитку безпроводних сенсорних мереж та мережного кодування даних належать іншим відомим науковцям, серед яких: С. Г. Бунін, В. О. Романов, В. А. Романюк, І. Акілдіз (I. Akyildiz), Р. Ахлсведе (R. Ahlswede), К. Фрагоулі (C. Fragouli) та ін.

Однак наявні праці у сфері обробки даних у системі залишкових класів та модулярній арифметиці не враховують ряд обмежень, які доцільно взяти до уваги при створенні безпроводних сенсорних мереж.

Таким чином, актуальною є науково-технічна проблема підвищення продуктивності та надійності функціонування безпроводних сенсорних мереж, вирішення якої зумовлює необхідність розробки комплексу методів, засобів і механізмів обробки даних на основі модулярної арифметики та системи залишкових класів.

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 “Безпроводні мультимедійні сенсорні мережі на основі модулярної арифметики та кодів Галуа для систем відеоспостереження” у межах програми двостороннього співробітництва між Україною та Китайською Народною Республікою у сфері науки і технологій (номер державної реєстрації 0113U004321);

– НДР на тему “Розробка теоретичних засад методів формування та цифрового опрацювання даних у розподілених спеціалізованих комп’ютерних системах” (номер державної реєстрації 0112U008458);

– НДР на тему “Розробка теорії та комп’ютерних засобів спеціалізованих комп’ютерних систем на основі теоретико-числових базисів Крестенсона – Галуа” (номер державної реєстрації 0106U012530);

– НДР на тему “Розробка теорії, методології побудови та технічних засобів спеціалізованих комп’ютерних систем” (номер державної реєстрації 0102U005764).

Мета і задачі дослідження. Мета дисертаційної роботи – підвищення ефективності функціонування безпроводних сенсорних мереж шляхом розвитку теоретичних основ та засобів перетворення, обробки і передачі даних.

Досягнення визначеної мети передбачає вирішення таких задач:

– проаналізувати сучасний стан та шляхи удосконалення методів обробки і передачі даних в безпроводних сенсорних мережах;

– розробити методи підвищення надійності передачі даних у безпроводних сенсорних мережах на основі модулярної арифметики;

– розробити метод мережного кодування даних на основі перетворення системи залишкових класів;

- розробити метод обробки та стиснення зображень у сенсорних мережах;
- створити концепцію побудови безпроводних сенсорних мереж на основі принципів колективного інтелекту;
- удосконалити метод передачі даних на основі розширення спектра сигналів методом стрибкоподібної зміни частоти;
- синтезувати нові й удосконалити відомі алгоритми і протоколи передачі даних у безпроводних сенсорних мережах;
- побудувати структури та реалізувати на програмованих логічних інтегральних схемах компоненти безпроводних сенсорних мереж на основі розроблених методів.

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

Предмет дослідження – методи та засоби перетворення, обробки і передачі даних у безпроводних сенсорних мережах на основі системи залишкових класів та модулярної арифметики.

Методи дослідження. При розв'язанні комплексу задач вирішуваної у дисертаційній роботі проблеми використовувались: теорія інформації та кодування, методи теорії чисел, методи завадостійкого кодування даних при розробці модулярних коректуючих кодів і методу мережного кодування; методи побудови розподілених комп'ютерних систем та принципи інтелекту рою при розробці концепції побудови безпроводних сенсорних мереж на основі колективного інтелекту; прикладна теорія цифрових автоматів, методи синтезу й аналізу цифрових схем при розробці пристроїв багаторівневого перетворення даних у системі залишкових класів; теорія та методи стиснення даних при розробці методу обробки зображень; теорія, методи і засоби проектування й імплементації цифрових систем на кристалі, мови опису апаратури при апаратній реалізації пристроїв перетворення та обробки даних.

Наукова новизна одержаних результатів:

1) вперше розроблено метод мережного кодування даних, який базується на перетворенні системи залишкових класів і забезпечує підвищення загальної пропускну здатності безпроводної сенсорної мережі за рахунок поділу пакетів даних на підпакети з використанням взаємно простих модулів різної розрядності та об'єднання пакетів даних у спільних вузлах для їхньої подальшої передачі;

2) вперше розроблено метод виправлення багатократних помилок на основі модулярних коректуючих кодів, який базується на розв'язанні системи модулярних рівнянь і забезпечує високу швидкодію та низьку обчислювальну складність процесів виявлення і виправлення помилок;

3) вперше розроблено метод контролю помилок при виконанні арифметичних операцій додавання, віднімання та множення даних на основі модулярних коректуючих кодів, який забезпечує виявлення і корекцію помилок при виконанні арифметичних операцій;

4) вперше розроблено метод перетворення та обробки зображень у системі залишкових класів, який забезпечує стиснення зображень без втрат і підвищення швидкодії процесу кодування / декодування за рахунок поділу зображення на частини та паралельної обробки окремих частин зображення;

5) розроблено концепцію побудови безпроводних сенсорних мереж на основі принципів колективного інтелекту та розподіленого процесора, яка, на відміну від відомих підходів, забезпечує виділення частини обчислювальних ресурсів безпроводних сенсорів для функціонування розподіленого процесора і високу живучість мережі за рахунок децентралізованого управління;

б) удосконалено модулярні коректуючі коди, які відрізняються від відомих тим, що обробляють дані, подані в позиційній системі числення, забезпечують високу швидкодію процесів обробки даних за рахунок паралельного виконання операцій модулярного множення та можливість адаптивної зміни коректуючої здатності кодів без зміни алгоритму роботи;

7) удосконалено метод передачі даних на основі стрибкоподібної зміни частоти, який базується на псевдовипадковій зміні частотного каналу і відрізняється від відомих методів тим, що забезпечує можливість виявлення помилок на фізичному рівні безпроводних сенсорних мереж за рахунок поділу даних у системі залишкових класів із розширеною системою модулів та передачі залишків на різних частотних каналах;

8) отримав подальший розвиток метод підвищення надійності передачі мультимедійних даних у безпроводних сенсорних мережах, який відрізняється від відомих методів можливістю виправлення пакетів помилок за рахунок формування транспонованої матриці даних, захищених модулярним коректуючим кодом, у кодері та виконання оберненої операції в декодері.

Практичне значення одержаних результатів полягає в розробці засобів підвищення надійності та корисної пропускну здатності безпроводних сенсорних мереж. Зокрема, практичну цінність мають такі результати:

1) структурні та схемотехнічні рішення пристроїв завадостійкого кодування / декодування даних на основі модулярних коректуючих кодів для виправлення однократних символічних помилок на ПЛІС, які характеризуються низькою апаратною складністю та високою швидкістю обробки даних;

2) синтезовано на ПЛІС пристрій декодування даних на основі модулярних коректуючих кодів для виправлення помилок в двох символах, в якому пошук мультиплікативних обернених елементів реалізовано у вигляді таблиць, що забезпечило підвищення швидкодії виправлення помилок;

3) створено генератор Verilog-коду для синтезу на ПЛІС декодера модулярних коректуючих кодів, який автоматично генерує Verilog-код за заданими параметрами, що дозволило підвищити швидкість розробки та уникнути помилок при написанні коду;

4) розроблено пристрій перетворення 12-розрядних даних 16 сенсорів, при цьому перетворення вхідних даних сенсорів, поданих у системі залишкових класів, у двійкову систему числення реалізовано на основі попарного об'єднання залишків за двома модулями, що дозволило зменшити розрядність даних у процесі виконання проміжних обчислень;

5) розроблено протокол передачі даних на основі запропонованого методу мережного кодування даних, що дозволило підвищити загальну пропускну здатність каналів зв'язку безпроводних сенсорних мереж.

Теоретичні та практичні результати дисертаційної роботи використано і впроваджено:

– на підприємстві ТОВ “Тернопільське конструкторське бюро радіозв'язку “Стріла” при розробці системи контролю електричних підстанцій з використанням технології безпроводних сенсорних мереж;

– на підприємстві ТОВ “Яваре” при розробці розподіленого програмного забезпечення систем моніторингу;

– у корпорації “Cypress Semiconductor” при розробці апаратно-програмного забезпечення модулів CY8CKIT-050 PSoC 5 і CyFi (CYRF7936) для підвищення надійності передачі даних у безпроводних сенсорних мережах;

– у КП “Тернопіль Інтеравіа” для підвищення ефективності роботи системи централізованого відеоспостереження міста Тернополя;

– при виконанні науково-дослідних робіт, які виконувались у Тернопільському національному економічному університеті;

– у навчальному процесі кафедр спеціалізованих комп'ютерних систем та інформаційно-обчислювальних систем і управління Тернопільського національного економічного університету при проведенні лекційних та лабораторних робіт з курсів “Проектування реконфігурованих апаратно-орієнтованих процесорів на програмованих логічних інтегральних схемах”, “Дослідження і проектування вбудованих КС”, “Системи передавання даних”, “Безпроводні комунікаційні системи”, а також при підготовці

магістерських робіт зі спеціальностей “Спеціалізовані комп’ютерні системи” та “Інформаційні управляючі системи і технології”.

Особистий внесок здобувача. Наукові положення, які містяться в дисертації, отримані здобувачем особисто. В роботах, опублікованих у співавторстві, здобувачу належать: [1, 30] – запропоновано метод виявлення та виправлення помилок на основі модулярних коректуючих кодів; [2, 51, 52] – запропоновано метод перетворення і стиснення зображень у системі залишкових класів; [3] – запропоновано метод обробки та передачі даних на основі системи залишкових класів та багатопрохідної маршрутизації; [5, 18, 34] – запропоновано модифікований метод стрибкоподібної зміни частоти та побудовано структуру передавача і приймача сигналів; [6] – запропоновано метод підвищення ефективності передачі даних у безпроводних сенсорних мережах; [7] – розроблено алгоритм та структуру пристрою виявлення і виправлення помилок на основі модулярних коректуючих кодів; [8, 38] – розроблено алгоритм виправлення помилок на основі двовимірних коректуючих модулярних кодів; [10, 27] – запропоновано концепцію побудови безпроводних сенсорних мереж на основі колективного інтелекту; [16, 33] – запропоновано нелінійний метод перетворення даних у безпроводних сенсорних мережах; [20] – запропоновано метод передачі даних у безпроводних оптичних каналах зв’язку; [21, 48] – розроблено спецпроцесор обробки даних у системі залишкових класів; [22, 23, 35, 54] – запропоновано метод завадостійкого кодування сигналів на фізичному рівні комп’ютерних мереж; [26] – розроблено структуру пристрою перетворення паралельного двійкового коду у код системи залишкових класів; [28, 39] – запропоновано підхід до виявлення багатократних символічних помилок на основі модулярних коректуючих кодів; [29] – побудовано структуру та визначено функції безпроводної мережі системи контролю аварійних ситуацій; [31, 32] – розроблено алгоритм пошуку оптимальних маршрутів для стаціонарних та мобільних вузлів на основі методу мурашиних колоній; [36] – розроблено алгоритм шифрування даних в системі залишкових класів; [37]

– проведено дослідження ефективності розділення каналів зв'язку в системі залишкових класів; [40] – розроблено структуру модуля безпроводної сенсорної мережі системи контролю лісових пожеж; [43] – здійснено реалізацію на ПЛІС кодера / декодера модулярних коректуючих кодів; [49, 50] – запропоновано метод мережного кодування в системі залишкових класів.

Апробація результатів дисертації. Результати роботи обговорювались і доповідались на міжнародних та всеукраїнських наукових конференціях, зокрема на: 3-rd, 5-th, 7-th, 8-th IEEE Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications: IDAACS'2005 (Sofia, Bulgaria, 2005), IDAACS'2009 (Rende-Cosenza, Italy, 2009), IDAACS'2013 (Berlin, Germany, 2013), IDAACS'2015 (Warszawa, Poland, 2015); 1st and 2nd IEEE International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems: IDAACS-SWS-2012, IDAACS-SWS-2014 (Offenburg, Germany, 2012, 2014); International Conference on Network Security, Wireless Communication and Trusted Computing, NSWCTC'2009, NSWCTC'2011, (Wuhan, China, 2009, 2011); International Conference Information Technology and Management Engineering "ITME 2011" (China, 2011); International Conference "Modern Problems of Radio Engineering, Telecommunications and Computer Science", TCSET (Lvov-Slavsko, 2002, 2012, 2014); 6-th International Conference Advanced Computer Systems and Networks: Design and Application. ACSN' 2013 (Lviv, Ukraine, 2013); 9-th International Scientific and Technical Conference Computer Science and Information Technologies, CSIT'2014 (Lviv, 2014); Міжнародній науково-практичній конференції "Сучасні інформаційні і електронні технології, СІЕТ" (м.Одеса, 2009, 2010, 2012, 2013); 5-й Міжнародній науково-практичній конференції "Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій" (м. Запоріжжя, 2010 р.); 4-й Науково-практичній конференції "Методи та засоби кодування, захисту й ущільнення інформації" (м. Вінниця,

2013); VI - й Українсько-польській науково-практичній конференції “Електроніка та інформаційні технології”, ЕЛІТ-2014 (сmt Чинадієво, Україна, 2014); Міжнародній конференції “Контроль і управління в складних системах” (м. Вінниця, 2012, 2014).

Публікації. За темою дисертації опубліковано 55 наукових праць, у тому числі: 22 статті в провідних фахових виданнях, з них 7 – у періодичних виданнях іноземних держав, 4 – у фахових виданнях України, які внесено до міжнародних наукометричних баз, 3 патенти України на винахід та 30 тез доповідей у збірниках матеріалів конференцій. 14 наукових праць індексовано в наукометричних базах *Web of Science* та *Scopus*.

Структура та обсяг дисертації. Дисертація складається із вступу, шести розділів, висновків, списку використаної літератури і додатків. Загальний обсяг роботи становить 327 сторінок, у тому числі 253 сторінки основного тексту, 117 рисунків, 27 таблиць та 7 додатків на 47 сторінках. Список літератури налічує 232 найменування.

РОЗДІЛ 1

СУЧАСНИЙ СТАН ТА ШЛЯХИ УДОСКОНАЛЕННЯ БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖ

1.1 Стан розробки безпроводних сенсорних мереж та їх застосування

Безпроводні сенсорні мережі (БСМ) – мережі розподілених мініатюрних автономних пристроїв, які використовуються для вимірювання, обробки та передачі даних про фізичні параметри та процеси навколишнього середовища [33, 34, 55, 56, 76, 77, 112, 113, 114, 115, 116, 117]. Реалізація БСМ стала можливою в результаті розвитку мікроелектроніки та досліджень на стику таких галузей, як вимірювання, обчислення та безпроводні комунікації (рисунок 1.1).



Рисунок 1.1 – Суміжні галузі досліджень БСМ

Дослідження в області БСМ інтенсивно проводяться в наступних напрямках [57, 109, 110, 113, 114, 123, 135, 136, 138, 140].

1. Схемотехніка:

- зменшення потужності споживання;
- розробка мікроелектромеханічних систем (MEMS);
- зменшення розмірів;
- зменшення вартості;

- використання відновлюваних джерел енергії.

2. Обчислення:

- обчислювальна потужність;
- вбудоване програмне забезпечення;
- обробка даних у вузлі;
- спільні методи обробки.

3. Мережні технології:

- самоконфігурація;
- масштабованість;
- динамічна топологія;
- управління топологією;
- гібридні мережі;
- розподілена маршрутизація і планування.

4. Безпроводний зв'язок:

- багатоінтервальна маршрутизація;
- енергоефективність;
- малий цикл роботи;
- ефективне управління доступом;
- комунікації в спільній частотній області.

На рисунку 1.2 наведено структурну схему безпроводного сенсорного вузла, який складається з шести основних компонентів: блоку сенсорів, аналогово-цифровий перетворювач (АЦП), процесора, пам'яті, радіомодуля та блоку живлення. Найбільш важливим компонентом безпроводного сенсорного вузла є блок обробки даних, який складається із процесора і пам'яті. Блок обробки даних здійснює приймання даних від АЦП, виконує арифметичні і логічні операції, зберігання даних та передачу даних на радіомодуль [117, 121].

Дослідженнями та розробкою апаратного і програмного забезпечення БСМ займаються ряд відомих університетів та провідні фірми на ринку електроніки, такі як Intel, IBM, Texas Instruments, Microchip, Freescale, NXP,

Atmel та інші [59, 14, 53, 131, 49]. Основні характеристики комерційних безпроводних модулів наведені в додатку А.

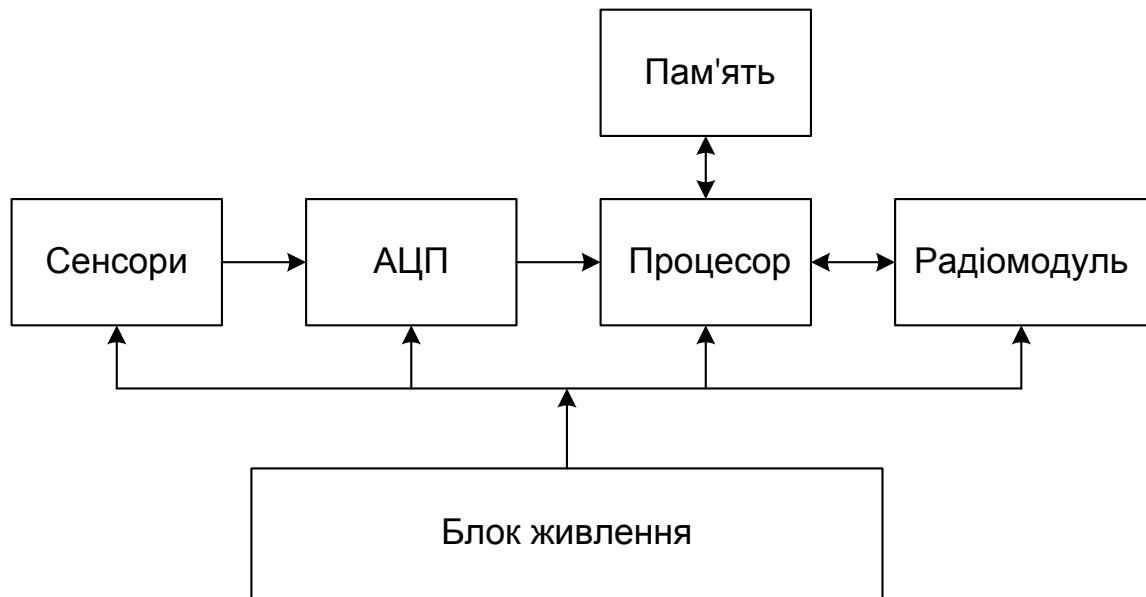


Рисунок 1.2 – Структурна схема безпроводного сенсорного вузла

На даний час успішно виконано ряд проектів з розробки та впровадження БСМ у різних галузях діяльності, наприклад, технічному і екологічному моніторингу, системах точного землеробства, цифровому зоопарку та інших [77, 113].

Разом з тим, при розробці БСМ дослідники стикаються з багатьма проблемами, в основному, викликаними відмовами передачі, обмеженими апаратними і обчислювальними ресурсами, живленням.

Безпроводні сенсорні мережі здатні збирати інформацію про стан фізичного середовища, виконувати просту обробку зібраних даних і передавати їх на віддалений сервер [4, 11, 113]. БСМ, які застосовуються в задачах моніторингу, найчастіше використовують топологію кластерного дерева (рисунок 1.3) [114]. В даній топології сусідні вузли на шляху до базової станції є як джерелами інформації, так і ретрансляторами даних інших вузлів. Як правило, базова станція здійснює передачу даних через безпроводний канал зв'язку (Wi-Fi, GSM або супутниковий канал) на віддалений сервер, підключений до мережі Інтернет.

На даний час, основою для розробки БСМ є стандарт IEEE 802.15.4 та стек протоколів ZigBee. Його розробником виступив альянс компаній Invensys, Honeywell, Mitsubishi Electric, Motorola, Philips та ін., який називається ZigBee [156].

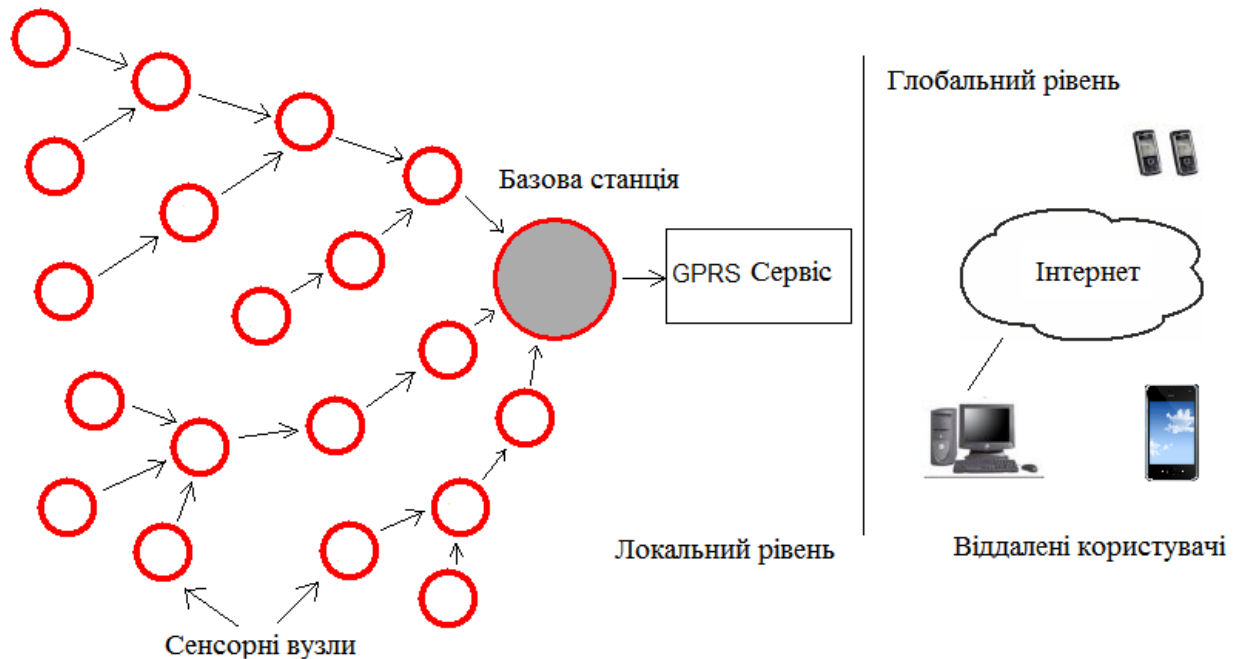


Рисунок 1.3 – Структура БСМ

Стек протоколів ZigBee побудований за принципом ієрархічної семирівневої моделі протоколів передачі даних у відкритих системах OSI (Open System Interconnection). Стек включає рівні стандарту IEEE 802.15.4, що відповідають за реалізацію каналу зв'язку, програмні мережні рівні і рівні підтримки додатків, визначені специфікацією альянсу ZigBee (рисунок 1.4) [156, 157, 232].

Передача даних на фізичному рівні сенсорних мереж здійснюється відповідно до стандарту IEEE 802.15.4 [156]. В якості середовища передачі даних використовується радіоканал або відкритий інфрачервоний канал [101, 156, 157].

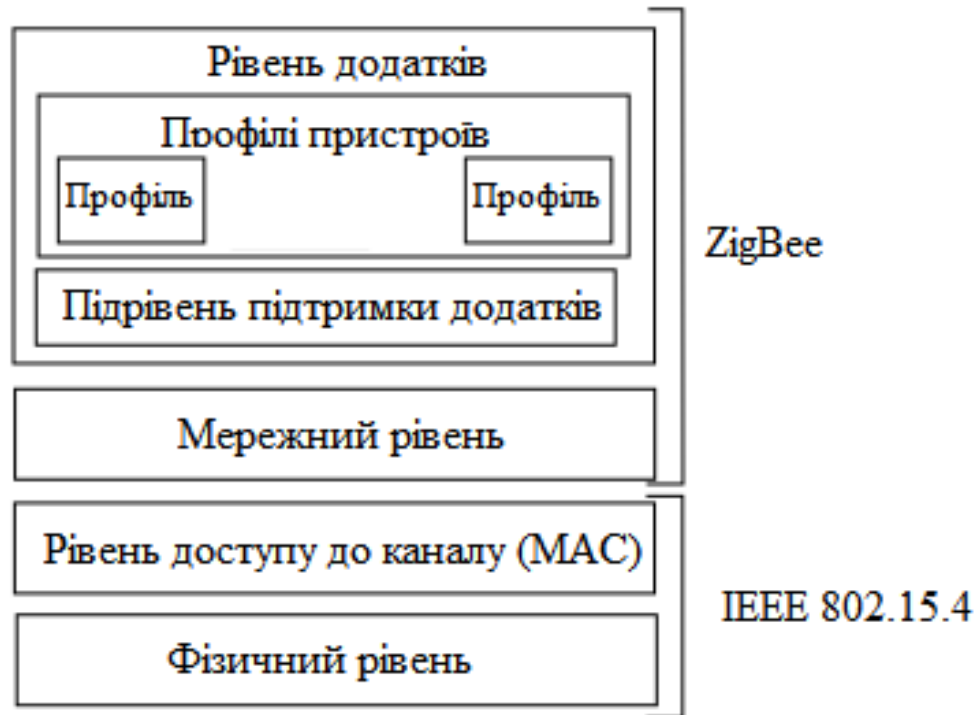


Рисунок 1.4 – Стек протоколів ZigBee

Стандарт IEEE 802.15.4 (ZigBee) передбачає роботу в трьох частотних діапазонах: один канал 868,0-868,6 МГц (для Європи); 10 каналів в діапазоні 902 - 928 МГц (крок центральних частот - 2 МГц); 16 каналів в діапазоні 2450 МГц (крок центральних частот - 5 МГц) (рисунок 1.5) [232].

Основною вимогою до вузлів БСМ є мінімальне споживання струму при забезпеченні заданої функціональності, що забезпечить тривалий час роботи від автономного джерела живлення. Одним з основних споживачів енергії вузла є радіотрансівер, отже, завдання зменшення енергозатрат при передачі даних є актуальною проблемою [170].

До безпроводних вузлів висуваються наступні вимоги:

- мінімальні розміри та вага;
- споживання мінімальної кількості енергії;
- робота з великою кількістю вузлів на обмеженій території;
- автономність і можливість роботи без обслуговування;
- адаптація до навколишнього середовища;
- низька вартість виробництва.

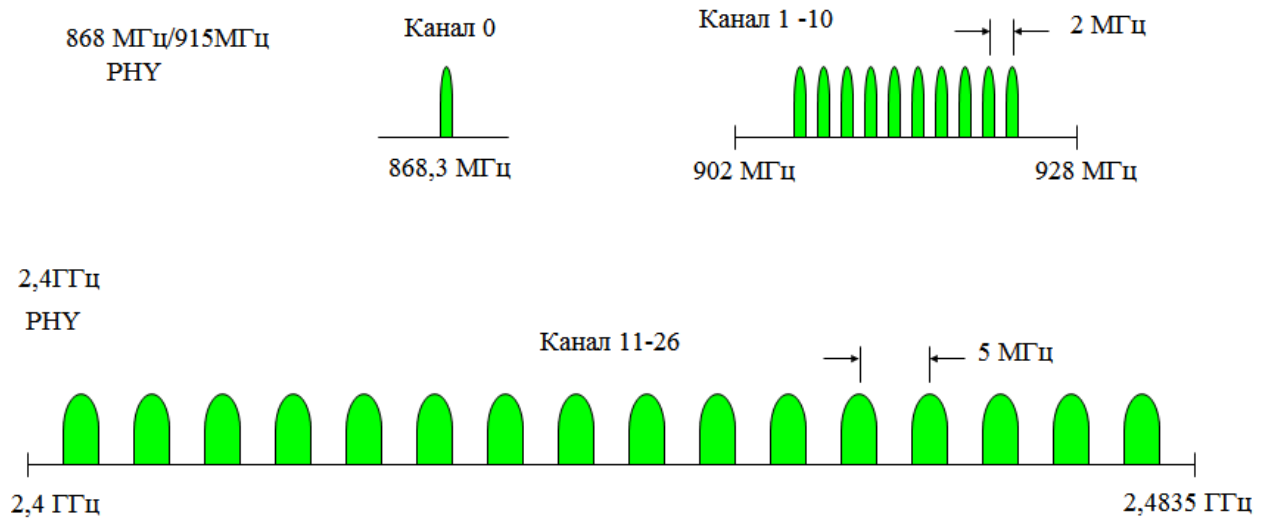


Рисунок 1.5 – Розподіл частотних каналів

Розроблена класифікація БСМ, яка включає технологію передачі, середовище передачі, тип вузлів, тип живлення, площу розгортання та область застосування (рисунок 1.6).

Технологія передачі. В якості середовища передачі в БСМ використовують радіоканал, безпроводний оптичний канал та канал на основі магнітної індукції [83, 85].

Середовище передачі. В залежності від середовища передачі даних розрізняють БСМ, які працюють на землі, під землею, під водою (див.рисунок 1.6) [228].

Наземні БСМ характеризуються великою кількістю недорогих вузлів (від сотень до тисяч), які передають дані на базову станцію. Оскільки джерело живлення обмежене і, як правило, не замінюється, вузли сенсорів можуть використовувати відновлювальні джерела енергії, наприклад сонячні батареї. Для збереження енергії вузли використовують багатоінтервальну маршрутизацію.

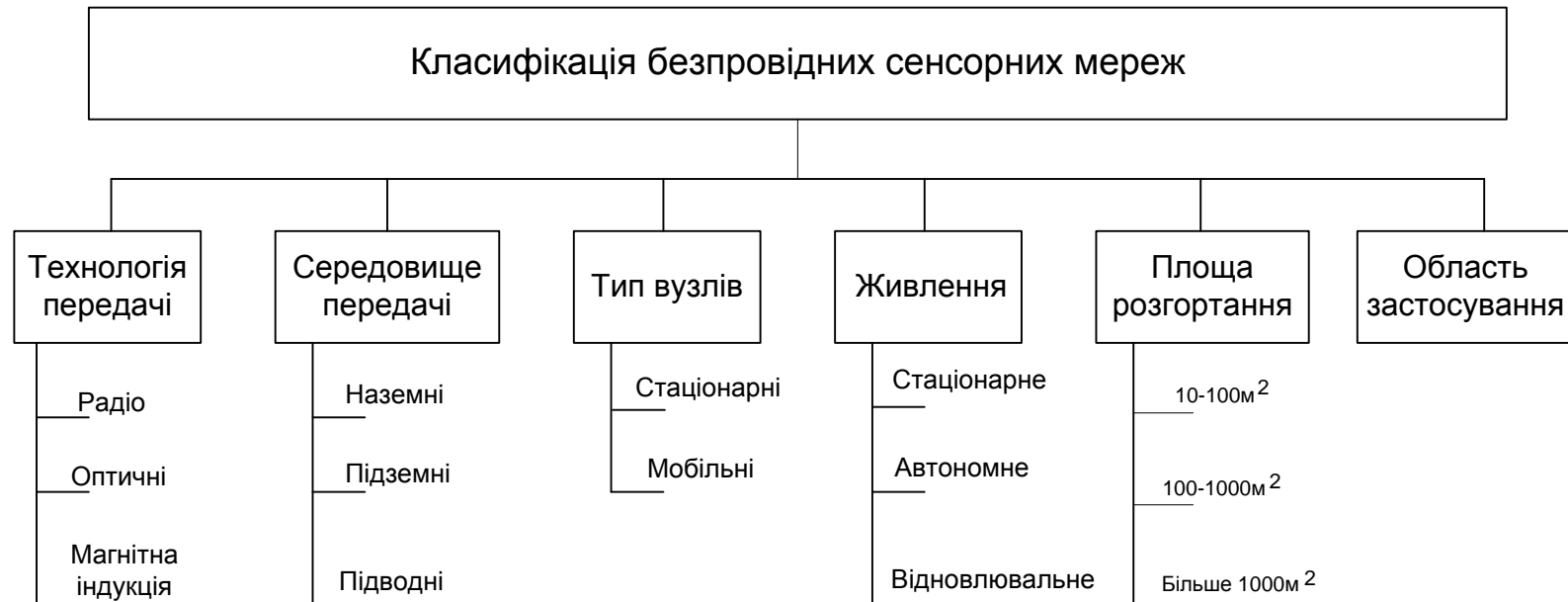


Рисунок 1.6 – Класифікація безпроводних сенсорних мереж

Підземні БСМ складаються з набору вузлів датчиків, розміщених під землею або в закритому просторі, такому як печери, шахти або метро [112, 118]. Основна відмінність між підземними і наземних БСМ – це середовище передачі, яке характеризується високим рівнем загасанням і втрати сигналу, що необхідно враховувати при розробці протоколів передачі даних.

Підводні БСМ складаються з датчиків, розміщених під водою, наприклад, для моніторингу морів та океанів [115]. Підводні мережі використовують акустичні хвилі і, відповідно, мають обмежену пропускну здатність, тривалу затримку поширення сигналу і проблеми завмирання сигналу. Підводні вузли повинні мати можливість самостійно адаптуватися до екстремальних умов навколишнього середовища океану. Підводні БСМ використовуються для моніторингу забруднення, сейсмічного моніторингу, моніторингу обладнання та підводної робототехніки [115].

Мобільні БСМ. Вузли мобільних БСМ можуть переміщатися і взаємодіяти з фізичним середовищем. При розробці БСМ з мобільними вузлами розробники стикаються з такими проблемами, як локалізація вузлів, навігація та управління рухомими вузлами, підтримка зв'язності мережі, мінімізація споживання енергії при русі, та розробка динамічних протоколів маршрутизації [32, 136, 138, 229, 61]. При розробці протоколів маршрутизації необхідно враховувати швидкість руху вузлів та максимальну відстань, на яку можуть змінитися координати вузла [32, 112].

1.1.1 Область застосування безпроводних сенсорних мереж

БСМ, в залежності від застосування, можуть використовувати різні типи сенсорів, зокрема, температури, вологості, руху, тиску, стану ґрунту, рівня шуму, диму, визначення хімічного складу (речовин, повітря), наявності або відсутності певних видів об'єктів, визначення швидкості, напрямку і розміру об'єкту, аудіо- та відеоданих [112]. Серед основних областей застосування БСМ необхідно відмітити наступні: екологічний та технічний

моніторинг, точне землеробство, охорона здоров'я, інтелектуальні будинки та системи безпеки [54, 113, 114, 115, 176].

Екологія. В екологічних додатках БСМ застосовуються для відстеження руху птахів, дрібних тварин і комах; моніторингу стану навколишнього середовища, параметри якого впливають на урожай і тваринництво; точного землеробства; біологічного моніторингу, моніторингу забруднення навколишнього середовища: повітря, водних ресурсів, в тому числі морів; виявлення лісових пожеж; метеорологічних або геофізичних досліджень [110, 114, 115, 226].

Охорона здоров'я: фізіологічний моніторинг (частота серцевих скорочень, кров'яний тиск, температура тіла, частота дихання, рівень стресу та інші параметри життєдіяльності); невідкладна допомога; моніторинг поведінки літніх людей [117, 119]; моніторинг персоналу і пацієнтів всередині лікарні [22, 115]. Фізіологічні дані, зібрані за допомогою сенсорних мереж, можуть зберігатися протягом тривалого періоду часу [108] і використовуватися для медичних досліджень [109].

Інтелектуальні будинки. З розвитком технологій безпроводні вузли будуть вмонтовані у побутові прилади, такі як пилососи, мікрохвильові печі, холодильники, люстри, мульти- та кавоварки та інші. Це дозволить побутовим приладам взаємодіяти один з одним і з зовнішньою мережею через мережу Інтернет або супутниковий канал зв'язку [110, 184].

Системи технічного моніторингу. БСМ ефективно використовуються в системах технічного моніторингу, зокрема для контролю мостів, нафтових танкерів та газопроводів; моніторингу будівель та споруд в сейсмічно активних районах. А також системи обліку енергоресурсів для індивідуального та промислового використання [117, 119, 120].

Системи безпеки. БСМ знаходять використання в індивідуальних та промислових системах безпеки, зокрема, системи контролю периметру, визначення вторгнення, віддаленого спостереження та інших [126].

В [66] запропонований спосіб оперативного переключення світлофорів з метою організації “зеленого коридору” для автомобілів спеціального призначення на основі технології безпроводних сенсорних мереж. Автомобілям спеціального призначення надається безпроводний пульт із запрограмованим кодом доступу. При в’їзді автомобіля в зону дії безпроводного модуля (0,1 – 1,0 км), який розміщений в світлофорі, передається код доступу за допомогою якого відбувається переключення світлофора в режим “увага” (жовтий миготливий колір) за визначеним алгоритмом. При виїзді із зони дії зв’язку, світлофор переходить в нормальний режим роботи (рис.1.7). Для виключення несанкціонованого доступу до керування світлофорами всі команди передаються у зашифрованому вигляді. Для захисту команд використано алгоритм 128 – бітного AES – шифрування та модифікований метод стрибкоподібної зміни частоти [89].

Безпроводні модулі, які розміщені на одному перехресті з’єднуються в мережу для коректного переключення світлофорів в зоні перехрестя (рис.1.8). Мережа організована на основі однотипних ZigBee модулів – маршрутизаторів, топологія мережі «зірка», або «mesh».

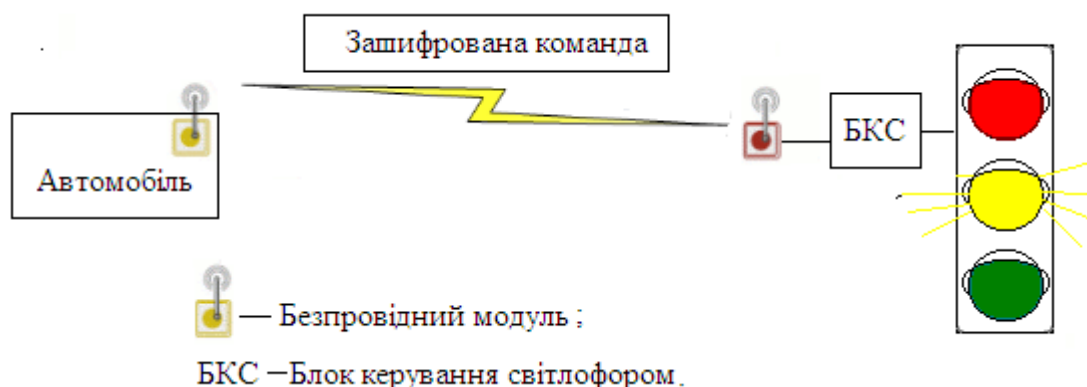


Рисунок 1.7 – Принцип безпроводного керування світлофором з використанням ZigBee модулів

В ZigBee модулях, які розміщені в світлофорі передбачена функція документування, яка зберігає в флеш пам’яті час та код пристрою з якого відбулося керування світлофором. При необхідності дані з флеш пам’яті

можна зчитати по безпроводному інтерфейсу. Дана мережа при підключенні відеосенсорів може бути використана для моніторингу дорожнього руху [94].

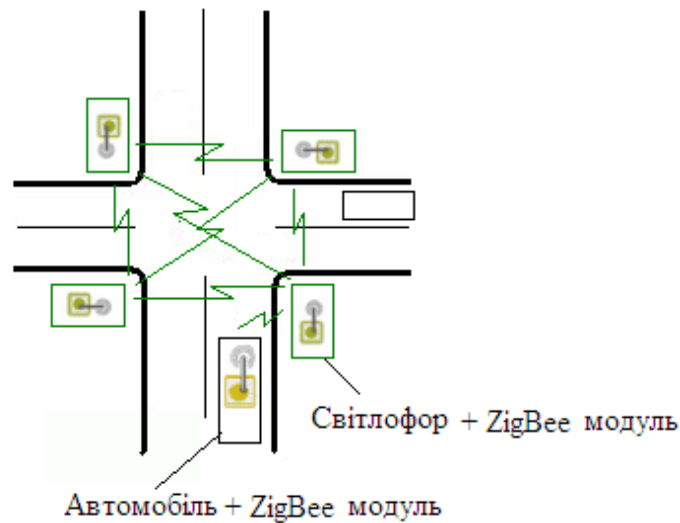


Рисунок 1.8 – Безпроводна сенсорна мережа керування світлофорами.

Даний підхід організації “зеленого коридору” дозволить підвищити швидкість та безпеку при проїзді регульованих перехресть автомобілями спеціального призначення, а також зменшить потребу у відключенні світлофорів на тривалий термін.

Для розширення функціональних можливостей системи моніторингу лісу в [226] запропоновано оснастити безпроводний вузол сенсором звуку (рис.1.9).

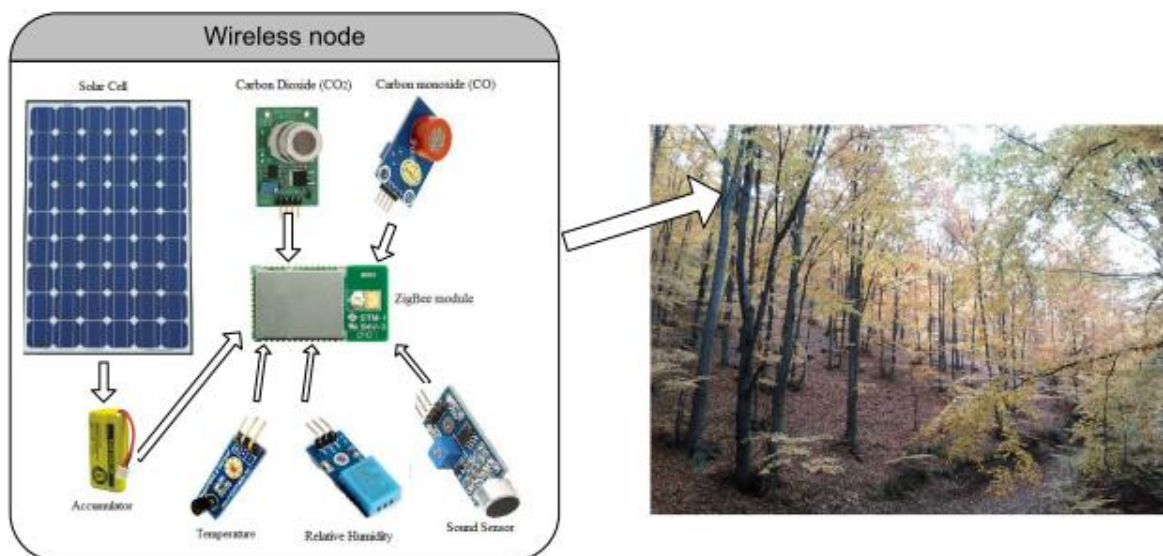


Рисунок 1.9 – Структура безпроводно вузла системи моніторингу лісу

Використання сенсору звуку дозволить підвищити достовірність виявлення події, а також розширить функціональні можливості системи моніторингу, зокрема забезпечить виявлення фактів несанкціонованого вирубування лісу в реальному часі.

1.1.2 Безпроводні мультимедійні сенсорні мережі

Наступним етапом еволюційного розвитку БСМ стали безпроводні мультимедійні сенсорні мережі (БМСМ), які містять аудіо та відео сенсори і здатні збирати, обробляти та передавати мультимедійні дані та/або оброблену інформацію про події на базову станцію (рисунок 1.10 а) [113]. Для зменшення обсягу даних безпроводний мультимедійний модуль (БММ) обладнаний детектором події та має можливість стиснення зображення. Безпроводний комунікаційний модуль (БКМ) здійснює ретрансляцію пакетів інших модулів на базову станцію (рисунок 1.10 б).

Проведений аналіз БМСМ показав, що при розробці архітектури мережі необхідно враховувати наступні параметри [163, 172]:

- Вимоги до якості обслуговування QoS для конкретних програм. На відміну від скалярних сенсорів, мультимедійні сенсори генерують фото і потоковий мультимедійний контент. Фотографії містять події, отримані протягом короткого періоду часу. Потоковий мультимедійний контент генерується протягом більш тривалих періодів часу і вимагає постійного режиму передачі.
- Обмеженість ресурсів. Вбудовані датчики мають обмеження стосовно запасу енергії, пам'яті, можливості обробки і достатньої швидкості передачі даних.

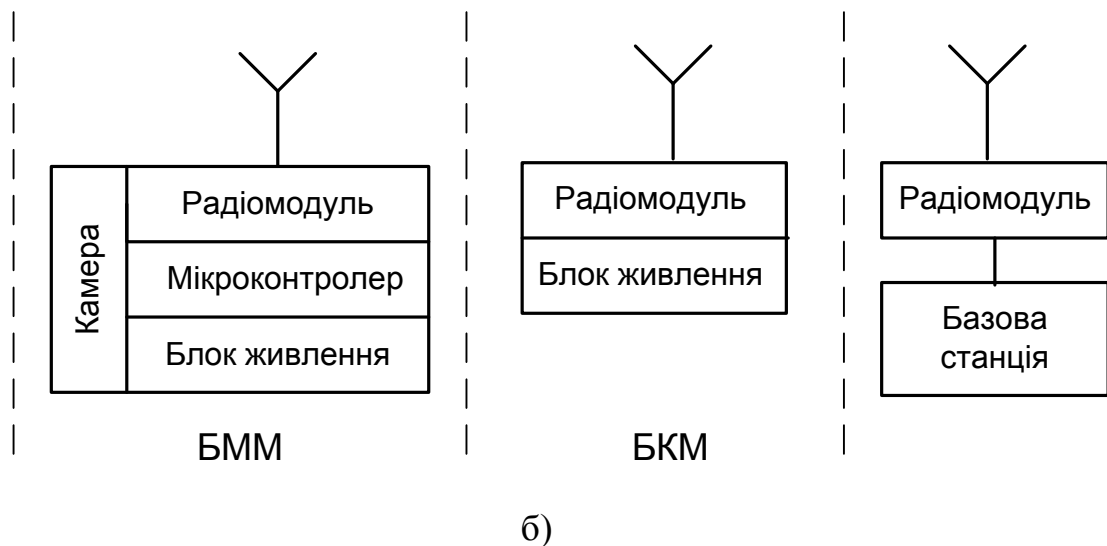
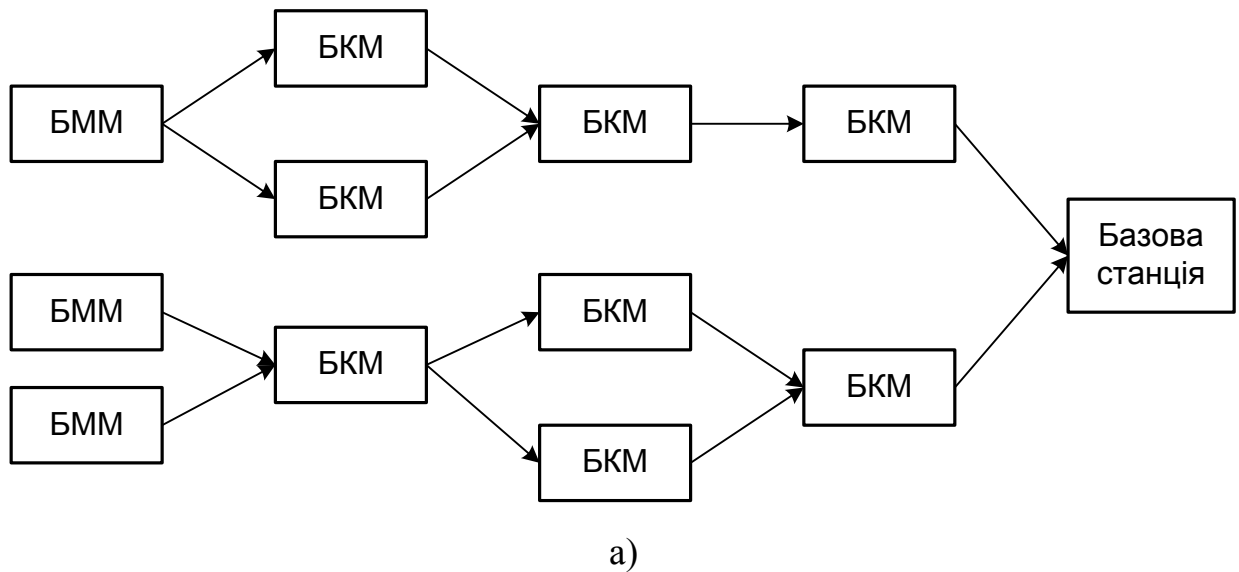


Рисунок 1.10 – Структура БМСМ: а) архітектура мережі; б) структура безпроводних вузлів. БKM – безпроводний комунікаційний модуль

- Змінна пропускна здатність каналів. У безпроводних сенсорних мережах пропускна здатність залежить від стану безпроводного каналу зв'язку між окремими сенсорами. Це, в свою чергу, залежить від взаємодії мережеских пристроїв та функцій, таких як управління потужністю, маршрутизації та інших. Таким чином, потужність і значення затримки на кожній ланці залежать від розташування вузла і безперервно змінюються, а також можуть мати пульсуючий характер, внаслідок чого забезпечення QoS є складним завданням.

- Крос-рівнева взаємодія функцій. Через загальну природу безпроводного каналу зв'язку в безпроводних сенсорних мережах з багатоінтервальною маршрутизацією існує чітка взаємозалежність між виконуваними функціями на всіх рівнях стека зв'язку. Ця взаємозалежність має бути обов'язково врахована при проектуванні протоколів зв'язку, спрямованих на забезпечення QoS.

- Широкопasmовий доступ. Мультимедійний контент, особливо відеопотоки, вимагає смугу пропускання, яка на порядок вища за доступну в даний час у БСМ. Наприклад, номінальна швидкість передачі даних в найсучасніших пристроях, які відповідають стандарту IEEE 802.15.4, становить 250 Кбіт/с. Необхідна швидкість передачі даних повинна бути значно вищою для потужних мультимедійних сенсорів при мінімальному енергоспоживанні. Таким чином, потрібно забезпечити високу швидкість та енергоефективні методи передачі даних. У цьому відношенні перспективною для БСМ є технологія передачі Ultra Wide Band (UWB) [163, 164].

- Методи обробки мультимедійних даних. Робота сучасних відеокодерів базується на наступних принципах: зменшення надлишковості в межах одного кадру; зменшення надлишковості між кадрами; кодування на основі прогнозування або оцінки руху. Впровадження сучасних методів інтелектуальної обробки передбачає наявність складних сенсорів, потужних алгоритмів обробки і, відповідно, потребує високого рівня споживання енергії.

- Мережева обробка мультимедійного контенту. Обробка та доставка мультимедійного контенту не є незалежними, і їх взаємодія має великий вплив на досягнення заданої QoS. Отже, необхідна незалежна від додатків архітектура мережі, яка б забезпечувала гнучку обробку мультимедійного контенту.

Основні вимоги до методів обробки даних в БСМ.

- Висока ефективність стиснення. Це необхідно для досягнення високого коефіцієнта стиснення, який дозволить зменшити вимоги до пропускної здатності каналів зв'язку та споживання енергії.
- Низька складність. Пристрої обробки мультимедійних даних вбудовані в безпроводні сенсори. Отже, вони повинні мати низький рівень складності для зниження витрат і споживати малу потужність для продовження терміну служби вузлів з автономним живленням.
- Завадостійкість. Методи обробки повинні забезпечувати надійне і стійке до впливу помилок кодування вихідних даних.

Традиційний підхід до обробки відеоданих ґрунтується на умові, що відео стискається один раз в кодері і декодується багато разів. Відповідно, кодер може бути складний, а складність декодера має бути низька. Даний принцип реалізований в широко використовуваних схемах ISO MPEG або ITU-T, рекомендаціях H.263 і H.264 (відомих також, як AVC або MPEG-4 Part 10), які зменшують швидкість передачі бітів за рахунок використання статистичних даних джерела. Оскільки для оцінки руху використовуються складні алгоритми обробки, які потребують високої обчислювальної складності, відповідно, ці методи вимагають наявності складних датчиків і призводять до високого рівня споживання енергії, в той час, як декодери є значно простішими. Для типових реалізацій сучасних стандартів стиснення відео, таких як MPEG або H.263 і H.264, кодер в п'ять-десять разів складніший за декодер [9, 13, 145, 67].

Для підвищення надійності передачі мультимедійних даних в [202] запропоновано розділяти пакети на частини в системі залишкових класів і передавати їх з використанням багатошляхової маршрутизації [169].

Отже, необхідно розробити простіші методи стиснення мультимедійних даних, які б забезпечували високу ефективність стиснення та мали низьку обчислювальну складність.

1.1.3 Аналіз області застосування безпроводних мультимедійних сенсорних мереж

Безпроводні мультимедійні сенсорні мережі мають значний потенціал, щоб стати основою для створення багатьох нових додатків. Вони можуть бути класифіковані наступним чином [113, 116, 171, 177]:

- Мультимедійні мережі відеоспостереження. Сенсорні мережі будуть використані для розширення та доповнення існуючих систем відеоспостереження для боротьби із злочинністю і терористичними актами. Мультимедійний контент, такий як відеопотоки і нерухомі зображення, разом з методами обробки сигналів і комп'ютерного зору, може бути використаний для пошуку зниклих безвісті людей або злочинців та виявляти і записувати інші потенційно небезпечні події (крадіжки, дорожньо-транспортні пригоди, порушення правил дорожнього руху та інші).

- Сенсорні мережі для підвищення енергоефективності житлових та промислових будівель. Поєднання відео, аудіо і скалярних сенсорів буде використовуватися в розумних будівлях для регулювати опалення, освітлення і кондиціонування повітря, зокрема, щоб зменшити загальний викид вуглекислого газу в повітря.

- Охорона здоров'я. Сенсорні мережі телемедицини можуть бути інтегровані з 3G/4G стільниковими мережами, щоб забезпечити широкий доступ до медичних послуг. Пацієнти матимуть медичні датчики для контролю таких параметрів, як температура тіла, артеріальний тиск, частота серцевих скорочень і робота дихальної системи. Дистанційні медичні центри стежитимуть за станом своїх пацієнтів, щоб попередити надзвичайні ситуації і оперативно надавати медичну допомогу в разі необхідності.

- Екологічний і структурний моніторинг. Мережа відеосенсорів використовується океанологами для визначення еволюції мілин за допомогою методів обробки зображень. Відео- та фотосенсори

використовуються також для контролю за станом мостів або інших цивільних споруд [116].

- Управління технологічними процесами. Мультимедійний контент, такий як зображення та звук, може бути використаний для критичних до часу систем керування технологічними процесами. При автоматизації виробничих процесів інтеграція систем машинного зору з БМСМ може спростити і додати гнучкості системам контролю для автоматизації виробництва.

Більшість потенційних застосувань БМСМ потребують переосмислення парадигми сенсорної мережі для створення механізмів отримання мультимедійного контенту з заданим рівнем якості обслуговування (QoS). Беручи до уваги, що головною проблемою при використанні сенсорних мереж було зведення до мінімуму споживання енергії, механізми для ефективної доставки додатків рівня QoS, такі як затримка і джиттер, не розглядалися в достатній мірі.

1.2 Переваги використання програмованих логічних інтегральних схем для побудови безпроводних мультимедійних сенсорних вузлів

Основою безпроводних вузлів на даний час є мікроконтролер, який виконує функції управління, обчислення, формування інтерфейсів для зв'язку з сенсорами та ін. В додатку А наведені основні характеристики мікроконтролерів, які використовуються в сучасних комерційних радіомодулях. Враховуючи обмежені обчислювальні можливості мікроконтролерів, програмовані логічні інтегральні схеми (ПЛІС) можуть стати хорошою альтернативою для побудови вузлів безпроводних сенсорних мереж [185, 121, 122, 51, 46]. ПЛІС являє собою електронний цифровий компонент, який використовується для створення реконфігурованих цифрових схем. На відміну від логічних елементів, які мають фіксовану

функцію, ПЛІС має невизначену функцію під час виготовлення. Перед використанням ПЛІС повинна бути запрограмована, тобто переконфігурована [24, 49, 50, 47, 19, 20].

Сучасні ПЛІС виготовляють за технологією 14 нм (Xilinx, Altera), а також випускаються оптимізовані платформи, які здатні працювати при ультранизькому енергоспоживанні, зокрема плати Actel IGLOO. Платформи на базі ПЛІС можуть надавати додаткові можливості обробки для типового сенсорного вузла зі зменшеним споживанням енергії. Крім того, ПЛІС забезпечують потужні можливості обробки, за рахунок арифметичних компонентів, таких як високошвидкісні помножувачі та суматори. Відповідно, ПЛІС можуть забезпечити шифрування, виконання високошвидкісних алгоритмів обробки даних, таких як маршрутизація, завадостійке кодування, стиснення даних та обробка зображень.

Використання реконфігурованих пристроїв для реалізації вбудованих систем з апаратними обмеженнями, таких як БМСМ, забезпечують додаткові переваги з точки зору зниження споживання енергії і підвищення обчислювальної потужності [167, 49].

В [216] показано, що при реалізації інструкцій програмного забезпечення в реконфігурованих апаратних засобах зменшення споживання енергії становить 35-70%, при цьому швидкодія зростає від трьох до семи разів.

Використання реконфігурованих апаратних засобів забезпечує також зменшення розміру і кількості компонентів, скорочує час виходу пристрою на ринок, підвищує гнучкість і здатність до модернізації [207, 49, 19].

На даний час мови опису апаратних засобів (HDL) стали основним інструментом для ефективною реалізації складних алгоритмів на ПЛІС. Мови опису апаратних засобів використовуються для опису роботи схеми, її проектування, тестування та для перевірки її роботи на етапі моделювання.

Переваги використання HDL [28, 158].

- **Можливість багаторазового використання.** Код, написаний на HDL в одному середовищі, може бути використаний в будь-якій системі, що підтримує дану мову. На відміну від HDL, використання схемних редакторів дозволяє повторне використання схеми за умови створення відповідних бібліотек і застосування того самого програмного середовища розробки. Модель поведінкового рівня HDL, як і більшість моделей рівня реєстрових передач (RTL), може бути повторно використана для декількох пристроїв.

- **Паралелізм.** В HDL паралелізм вбудований в мову, в порівнянні зі звичайним програмним забезпеченням, що використовують для імітації апаратного забезпечення послідовні інструкції виконуються послідовно.

- **Часові параметри схеми.** За допомогою відповідних інструментів моделювання розробнику доступна вся інформація про часові параметри розроблюваної схеми, які також залежать від мікросхем конкретного виробника.

- **Оптимізація.** Метод проектування ПЛІС включає в себе написання рівнів RTL, а потім використовується для синтезу пристрою за допомогою відповідного програмного забезпечення. Процес синтезу полягає у створенні еквівалентного опису HDL рівня елементів, який потім може бути оптимізований відповідно до конкретного обладнання. Модель рівня RTL можна записати і синтезувати для певного типу ПЛІС.

- **Стандарти.** Мови Verilog і VHDL є стандартизовані. Тому пристрій, описаний даними мовами, буде коректно працювати в будь-якому іншому програмному середовищі, яке їх підтримує.

Файли конфігурації ПЛІС генеруються за допомогою систем автоматизованого проектування, таких як Quartus II фірми “Altera”, Xilinx ISE фірми “Xilinx” та ін. Це програмне забезпечення надає все необхідне для розробки, модифікації і генерування конфігурації ПЛІС з використанням VHDL, Verilog, схематичних, графічних редакторів або методології проектування на основі блоків (рисунок 1.11).

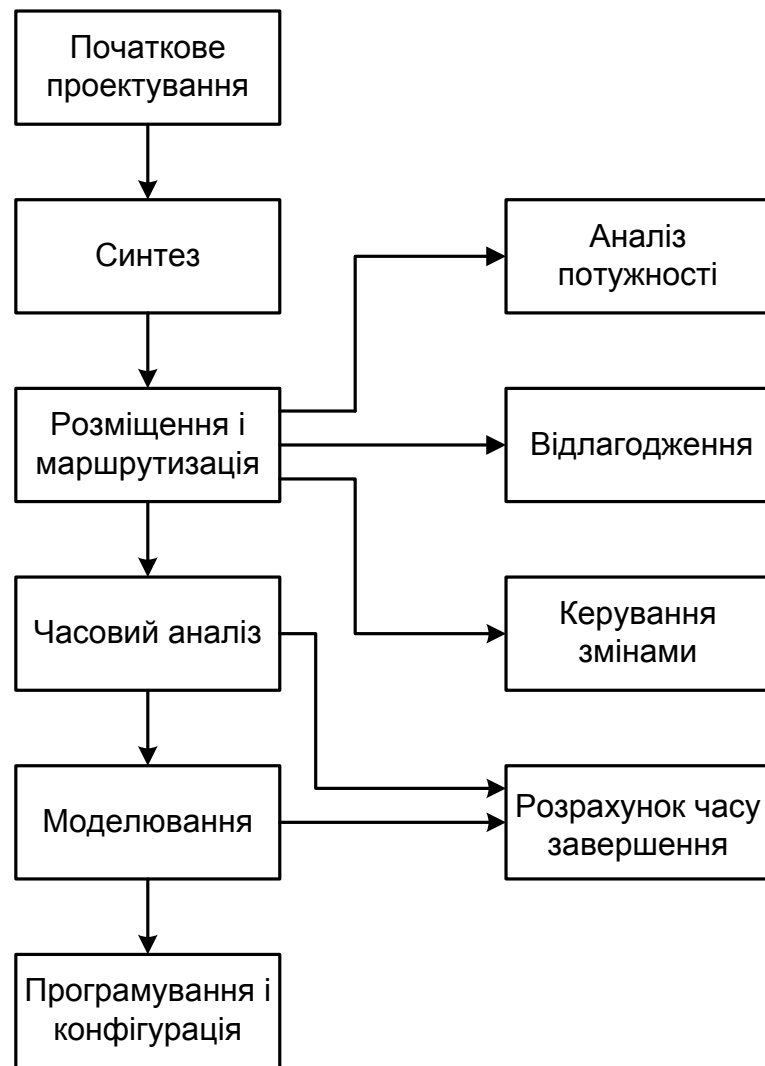


Рисунок 1.11 – Послідовність етапів розробки проекту на ПЛІС з використанням програмного забезпечення Quartus II.

Початкове проектування включає в себе розробку пристрою на основі мов VHDL, Verilog або графічних редакторів. Основними етапами обробки HDL є процес моделювання і синтезу. Процес моделювання інтерпретує висловлювання HDL з метою отримання часових діаграм, які показують поведінку апаратного пристрою ще до його імплементації в ПЛІС. Моделювання дозволяє виявити функціональні помилки в схемі без необхідності виготовлення пристрою. Розрахунок часу завершення оптимізує час відповідно до мети синхронізації шляхом управління синтезом і функціями розміщення і оптимізації.

Варто відзначити можливість динамічної реконфігурації сучасних ПЛІС. Динамічна реконфігурація дозволяє змінювати систему під час роботи, тобто змінюється зміст логічних блоків, а також конфігурація з'єднань. Це дозволяє переналаштовувати деяку частину або всю ПЛІС. Динамічна реконфігурація дозволяє звести до мінімуму площу схеми за допомогою реконфігурації послідовних частин алгоритму, завантажуючи за один раз один крок алгоритму, та знизити споживання енергії за рахунок покрокового виконання алгоритму [121].

В більшості випадків БСМ проектуються з заздалегідь визначеними функціями, однак часто виникає ситуація, що призводить до необхідності розширення функцій мережі або навіть зміни завдання. При реалізації вузлів на заказних інтегральних схемах здійснити реконфігурацію вузлів неможливо.

Для усунення цього недоліку в [147] були запропоновано три способи реконфігурації вузлів, побудованих на ПЛІС з використанням програмних процесорів. Перший статичний спосіб передбачає, що програмне забезпечення для реконфігурації вузлів було попередньо запрограмовано в системі і при виконанні певних умов система буде переналаштована відповідно до закладеного алгоритму. Другий псевдо - статичний спосіб для зміни конфігурації системи використовує безпроводний зв'язок. Обидва способи вимагають припинення всіх операцій для реконфігурації вузлів. Третій спосіб – динамічна реконфігурація, яка дозволяє часткову реконфігурацію системи через безпроводний зв'язок, щоб мінімізувати вплив зміни апаратних засобів на функціонування сенсорних вузлів.

Серед переваг використання ПЛІС, як обчислювального середовища сенсорного вузла, необхідно виділити наступні [121]:

– обчислювальна ефективність. У порівнянні з мікроконтролерами обробка даних в ПЛІС може відбуватися паралельно, відповідно зростає обчислювальна ефективність;

– енергоефективність. Енергетична ефективність програмного ядра процесора визначається, як відношення споживання енергії до його обчислювальної ефективності. Обчислювальну ефективність ПЛІС можна регулювати за допомогою тактового генератора;

– реконфігурація. Використовуючи цю функцію, програмні процесори сенсорних вузлів можуть бути оновлені або повністю реконфігуровані віддалено через безпроводний канал зв'язку;

– обчислювальна потужність. У багатьох випадках мікроконтролери не можуть забезпечити обчислювальну потужність, необхідну для виконання складних алгоритмів.

Основним недоліком використання ПЛІС для БСМ з автономним живленням залишається їх високе енергоспоживання. Однак, усвідомлюючи необхідність використання малопотужних ПЛІС для ряду застосувань, великі виробники, такі як “Altera” і “Xilinx”, почали виробництво нового ряду малопотужних високопродуктивних ПЛІС, таких як Altera Stratix-V і Xilinx Artix®-7.

Беручи до уваги сильні сторони мікроконтролерів та ПЛІС, вузли БСМ можуть бути реалізовані на основі [167]:

- 1) комбінації мікроконтролера та ПЛІС;
- 2) ПЛІС з процесорним ядром всередині;
- 3) ПЛІС без процесорного ядра всередині.

Наведені варіанти реалізації вузлів БСМ мають значні переваги, які забезпечує висока обчислювальна потужність ПЛІС в поєднанні з гнучким програмним забезпеченням мікроконтролера.

Застосування ПЛІС робить процес розробки більш гнучким, оскільки дає можливість використовувати готові програмні ядра мікропроцесорів з необхідним набором периферійних контролерів.

Отже, можливість реконфігурації, легкість інтеграції периферійних пристроїв і висока обчислювальна потужність надає ПЛІС значні переваги

для використання їх в якості апаратних платформ безпроводних мультимедійних сенсорних мереж.

1.3 Аналіз методів підвищення корисної пропускної здатності безпроводних сенсорних мереж

Враховуючи низьку швидкість передачі даних в БСМ (250 кбіт/с), актуальною є проблема розробки методів обробки та передачі даних, орієнтованих на ефективне використання та підвищення корисної пропускної здатності каналів зв'язку БСМ [156, 193, 196].

Ефективного використання корисної пропускної здатності БСМ можна досягти за рахунок оптимізації структури мережевих протоколів та попередньої обробки даних сенсорів у безпроводних вузлах (методи стиснення, агрегування та фільтрації даних) [70, 163, 164, 188].

Для підвищення загальної пропускної здатності каналів зв'язку БСМ використовують методи мережного кодування та багатошляхової маршрутизації [5, 109, 124, 125, 138, 141, 142, 144, 143].

Зменшити відсоток службових даних в структурі мережевих протоколів можна за рахунок передачі повідомлень максимальної довжини. Однак, системи моніторингу та управління, побудовані на основі технології БСМ, характеризуються малою довжиною інформаційних повідомлень. Наприклад, обсяг даних сенсорів при вимірюванні фізичних величин (температури, тиску, вологості, радіаційного фону та ін.) складає лише 12 – 32 біти. Збільшити розмір повідомлення в БСМ можна об'єднавши дані різних сенсорів в один пакет. Одним із способів збільшення розміру пакету (об'єднання даних сенсорів) є конкатенація (Concatenation) даних [69]. Якщо $\alpha = a_1 \dots a_n$ і $\beta = b_1 \dots b_m$ слова в алфавіті A , то конкатенацією слів α і β буде слово γ в тому ж алфавіті A , що визначається $\gamma = \alpha \beta = a_1 \dots a_n b_1 \dots b_m$.

Однак, якщо виміряні значення сенсорів мають різну розрядність, використання відомих методів конкатенації призводить до збільшення надлишковості [31]. Для зменшення трафіку від сенсорних вузлів використовують різні підходи. Так, в роботах [149, 174] пропонується технологія «мобільних агентів» - програмного коду, при передачі якого вихідний обсяг даних може бути зменшений за допомогою усунення надлишковості детермінованими методами. Але такий підхід значно обмежує сферу застосування даної технології через обмеження на архітектуру мережі, відстані до ядра (кластера) та інші.

Ще одним підходом до зменшення кількості даних, які передаються на базову станцію, є методи агрегування даних. Агрегування даних зазвичай виконується шляхом передавання тільки максимальних, мінімальних та середніх значень виміряних даних [135, 140]. Це зменшує кількість даних, які передаються, і, відповідно, зменшує енергоспоживання вузлів. Тим не менше, цей метод втрачає структуру даних, оскільки він забезпечує передачу тільки статистичних даних без їх миттєвих значень [148]. Деякі автори частково вирішують цю проблему використанням адаптивного агрегування, яке враховує розташування вузлів, динаміку зміни даних та пропускну здатність мережі [108, 114, 130]. Проте ці методи призводять до втрати первинної структури даних при високому ступені агрегації. Отже, використання методів агрегування даних є виправданим при низькій динаміці зміни даних сенсорів та високій щільності розміщення вузлів в сенсорній мережі.

В [38] пропонується оптимізація трафіка шляхом декореляції даних, тобто усунення надлишковості. Однак використання даного підходу можливе тільки для потоків даних близько розміщених вузлів та при вимірюванні однакових фізичні величини, що також суттєво обмежує його застосування.

Найбільш універсальним підходом до зменшення трафіку є використання методів ущільнення даних, яким в останні роки приділяться

значна увага [9, 13, 48, 163, 164, 172, 197, 211]. Використання методів ущільнення даних в вузлах БСМ дозволяє значно скоротити обсяг даних, які необхідно передавати, і, тим самим, підвищити тривалість роботи мережі [163].

Проте даний підхід є малоефективним при обробці даних невеликого обсягу (2-3 байти), а також призводить до збільшення затримки передачі даних, що обмежує його використання в системах моніторингу реального часу. Як відомо, використання алгоритмів ущільнення при обробці малих обсягів даних призводить до зворотного ефекту, тобто до збільшення обсягу вихідних даних за рахунок введення додаткових даних для декодування повідомлень [9]. Ще один фактор, який ускладнює використання методів стиснення даних в БСМ, – обмежені апаратні ресурси сенсорних вузлів. Так, проведені в [211] дослідження показують, що відомі алгоритми, які забезпечують високий коефіцієнт стиснення, наприклад, алгоритми на основі методів прогнозування, арифметичного кодування, вимагають більше затрат енергії, ніж система без ущільнення. Це пов'язано зі складністю реалізації алгоритмів стиснення даних, які вимагають значної обчислювальної потужності та пам'яті.

1.3.1 Методи мережного кодування

Теорія кодування виділяє три основних типи кодування: кодування джерела, каналне кодування та мережне кодування [63, 111]. Перший тип спрямований на ущільнення даних в джерелі, другий – вводить надлишкові біти на каналному рівні для підвищення надійності передачі даних. Кодування третього типу відбувається в проміжних вузлах мережі та на різних рівнях мережевого стеку.

Пакети даних в сучасних комп'ютерних мережах використовують спільні ресурси, але інформація в них не змішується. Даний принцип використовує мережне кодування (network coding) – новий напрямок в теорії

інформації, який дозволяє підвищити ефективну пропускну здатність каналів зв'язку комп'ютерних мереж без значної зміни інфраструктури [58, 92, 114, 124, 129, 142, 151, 161, 178, 195, 203, 230].

Методи мережного кодування почали активно розвиватися з роботи Ahlswede R., в якій показано, що використання мережного кодування дає змогу досягти загальної швидкості передачі близької до максимальної [173, 203].

Основною проблемою мережного кодування є вибір методу комбінування пакетів. На даний час мережне кодування розвивається в двох теоретичних напрямках: лінійне кодування та кодування на основі китайської теореми про залишки [161].

Основний принцип мережного кодування полягає в тому, що проміжний вузол об'єднує декілька прийнятих пакетів і передає їх наступним вузлам мережі [195, 203].

Розглянемо принцип мережного кодування на прикладі обміну пакетами в безпроводній мережі з трьома вузлами. Нехай вузли А, В обмінюються повідомленнями X_1 , X_2 через вузол (ретранслятор) С (рисунок 1.12) [92].

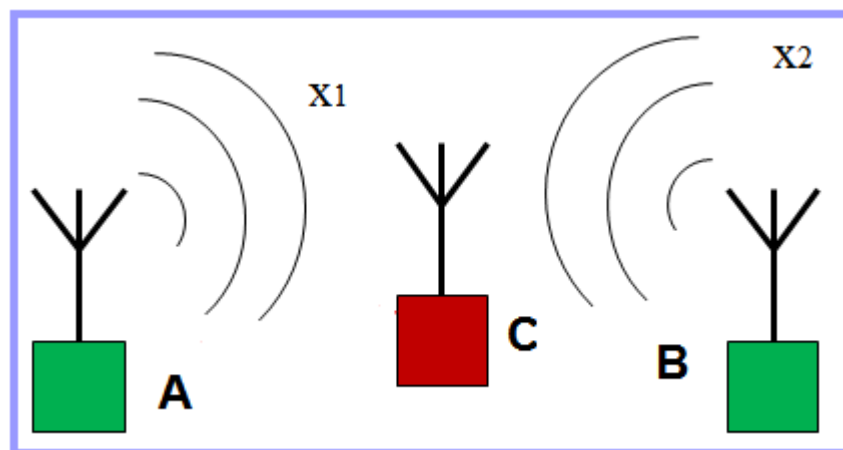


Рисунок 1.12 – Безпроводна мережа з трьох вузлів

Для безпроводної мережі без кодування протокол передачі має вигляд (рисунок 1.13):

$$t1 \quad x_1 : A \rightarrow C;$$

$$t2 \quad x_2 : B \rightarrow C;$$

$$t3 \quad x_2 : C \rightarrow A;$$

$$t4 \quad x_1 : C \rightarrow B.$$

Отже, для обміну повідомленнями в мережі без кодування між вузлами А і В необхідно чотири кадри.

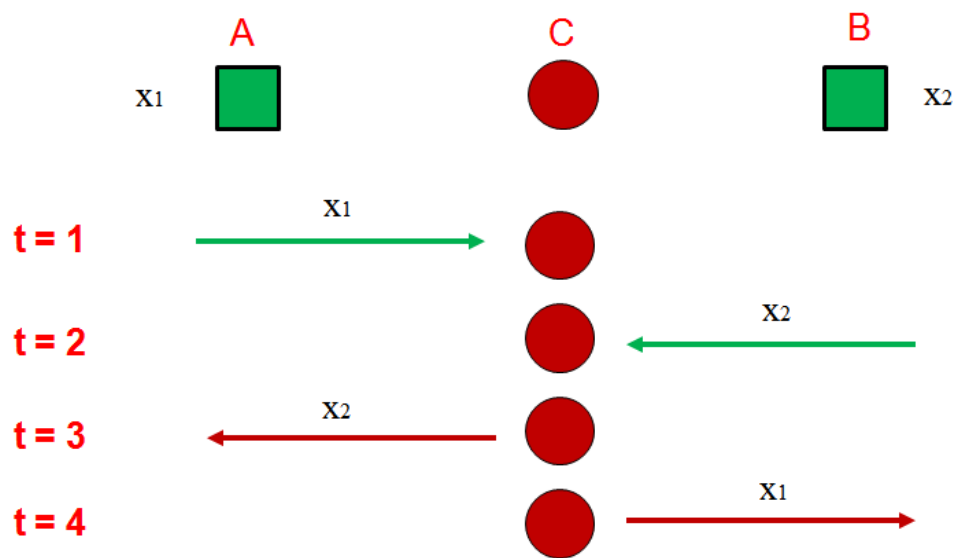


Рисунок 1.3 – Схема передачі без мережного кодування

При використанні мережного кодування ретранслятор зберігає пакети і утворює їх лінійні комбінації. В даному випадку протокол передачі буде мати вигляд (рисунок 1.14):

$$t1 - x_1 : A \rightarrow C;$$

$$t2 - x_2 : B \rightarrow C;$$

$$t3 - x_1 \oplus x_2 : C \rightarrow A; C \rightarrow B.$$

В кадрі 3 вузол S формує суму прийнятих пакетів за модулем 2 і передає пакет $x_1 \oplus x_2$ (рисунок 1.14). Вузли А і В приймають цей пакет і обчислюють необхідні пакети:

$$x_2 = (x_1 \oplus x_2) \oplus x_1;$$

$$x_1 = (x_1 \oplus x_2) \oplus x_2.$$

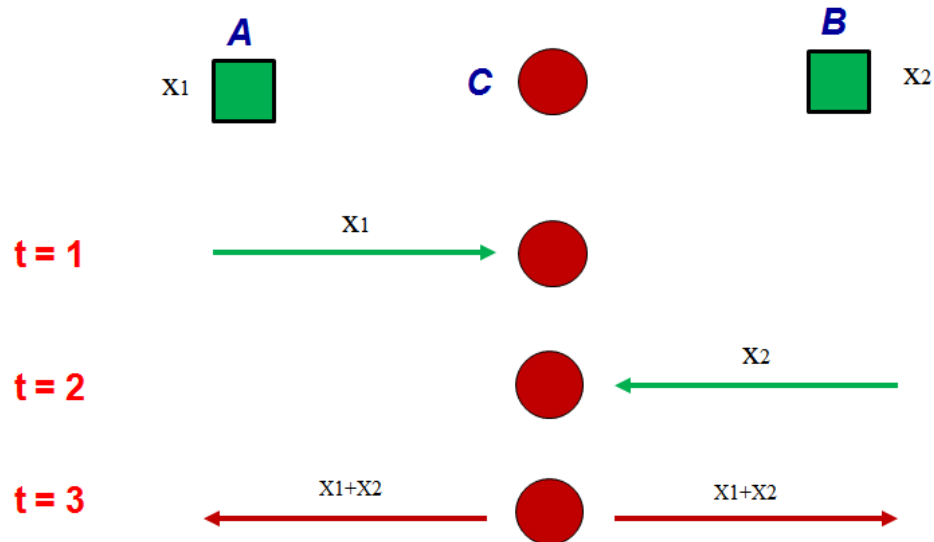


Рисунок 1.14 – Передача пакетів з використанням мережного кодування

Отже, при використанні мережного кодування для обміну повідомленнями між вузлами А і В необхідно 3 кадри.

Використання мережного кодування дозволяє збільшити пропускну здатність та підвищити надійність мережі [142].

При мережному кодуванні кожний вихідний пакет, що передається по мережі, є лінійною комбінацією отриманих пакетів, де операція додавання виконується над полем $GF(2^s)$ [41, 44, 141]:

$$X = \sum_{i=1}^n g_i \cdot H^i,$$

де H^i – вихідні пакети одного або декількох джерел інформації;

g_i – коефіцієнти поля $GF(2^s)$, символ g_i складається із s бітів.

Для того, щоб відновити закодовані пакети $(g_1, H^1), \dots, (g_m, H^m)$, необхідно розв'язати систему m рівнянь

$$\left\{ X^j = \sum_{i=1}^n g_i^j \cdot H^i \right\}, j = \{1, \dots, m\},$$

в якій H^i є невідомими. Для розв'язання системи m рівнянь з n невідомими необхідно, щоб $m \geq n$. Однією з важливих задач при використанні мережного кодування є вибір кожним вузлом мережі лінійних комбінацій. Найпростіший алгоритм – кожний вузол довільно, незалежно і децентралізовано, з рівною ймовірністю вибирає елементи поля $GF(2^s)$. В [142] розглянуто різні підходи до вибору коефіцієнтів кодування та алгоритмів декодування.

Пізніше були розроблені лінійні схеми кодування для мереж з багатоадресною передачею [142, 161]. Перевагою схем випадкового лінійного кодування є те, що вони можуть бути використані для мереж з динамічною топологією [142]. Проте цей підхід зменшує корисну пропускну здатність за рахунок додавання до кожного пакету заголовку, в якому зберігаються коефіцієнти кодування. Даний підхід прийнятний для комп'ютерних мереж, в яких розмір пакетів великий (1500 байт). Однак для безпроводних сенсорних мереж, де пакети набагато менші (127 байт) і пропускну здатність обмежена, він є надлишковим.

В [195] запропоновані схеми комбінування втрачених пакетів з різних приймачів таким чином, що декілька приймачів можуть відновити свої втрачені пакети. Це забезпечує підвищення пропускну здатності мережі за рахунок зменшення кількості повторних передач в безпроводних мережах.

В [58, 161] показано, що коди з малою щільністю перевірок на парність (Low Density Parity Check Codes), LDPC - коди дозволяють вирішити завдання, що виникають при реалізації мережного кодування. Однак для їх ефективного застосування в безпроводних мережах необхідно модифікувати алгоритми передачі і декодування повідомлень [142].

Розроблені методи лінійного мережного кодування призводять до значного збільшення службових даних в пакеті протоколу і, відповідно, не можуть бути використані в БСМ, які характеризуються малими розмірами пакетів і мають обмежену пропускну здатність каналів зв'язку [129].

На відміну від методів лінійного мережного кодування, мережне кодування на основі китайської теореми про залишки дозволяє зменшити обчислювальні затрати на кодування даних та обсяги службових даних в протоколах передачі даних [124, 129, 141].

В [141] показано переваги мережного кодування в системі залишкових класів при передачі даних «один до багатьох». В [129] запропоновано використовувати множину взаємно простих модулів для кодування і передачі повідомлень на основі китайської теореми про залишки на прикладі багатоадресної передачі для топології мережі «метелик». Однак дана схема є надлишковою для БСМ, оскільки відбувається дублювання залишків за різними маршрутами, що приводить до зменшення корисної пропускної здатності мережі та збільшення енергозатрат.

В БСМ безпроводні вузли, зазвичай передають виміряні дані на базову станцію, тобто використовується принцип передачі «багато до одного».

Мережне кодування в СЗК забезпечує підвищення продуктивності мережі за рахунок зменшення кількості пакетів, які необхідно передавати. Проте методи підвищення корисної пропускної здатності БСМ при мережному кодуванні в СЗК ще потребують дослідження.

Враховуючи вище сказане, відкритою залишається проблема розробки методів мережного кодування в системі залишкових класів, які враховують особливості топології БСМ.

1.4 Аналіз підходів підвищення надійності передачі даних у безпроводних сенсорних мережах

При забезпеченні необхідного рівня надійності роботи БСМ, на відміну від безпроводних комп'ютерних мереж, необхідно враховувати обмеження, які накладаються низькими обчислювальними ресурсами вузлів

та використанням автономного живлення [27]. 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, зростають вимоги до надійності передачі кожного пакету. Отже, для ефективної роботи БСМ каналний рівень повинен забезпечувати надійну і енергоефективну передачу даних при різних режимах роботи (топологіях) мережі.

Основним механізмом підвищення надійності передачі даних у БСМ є використання схем контролю помилок. Основною метою схем контролю помилок є забезпечення надійного зв'язку в безпроводному каналі, в якому помилки виникають через перешкоди, завмирання та втрати бітової синхронізації [168]. Це призводить до каналних помилок, які впливають на цілісність пакетів, що відправляють сенсорні вузли. До збільшення помилок також призводить низька потужність передавачів у БСМ, що пов'язано з використанням автономного живлення.

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

В БСМ використовують наступні схеми контролю помилок [21, 102, 132, 231]:

- 1) управління потужністю;
- 2) автоматичний запит повторної передачі (ARQ);
- 3) пряме виправлення помилок (FEC);
- 4) гібридна ARQ (HARQ).

1.4.1 Управління потужністю

Більшість трансиверів (JN5148, JN5161, CC2420), які використовуються в вузлах БСМ, підтримують до чотирьох рівнів потужності передачі, що можуть встановлюватися і адаптивно змінюватися в залежності від стану каналу зв'язку. Встановлення більшої потужності трансивера зменшує частоту помилок шляхом покращення співвідношення сигнал-шум, однак збільшує споживання енергії. Крім того, реалізація механізмів управління потужністю ускладнює протоколи обміну даними та потребує наявності додаткових ресурсів мікроконтролера та обсягів пам'яті (рисунок 1.15) [208].

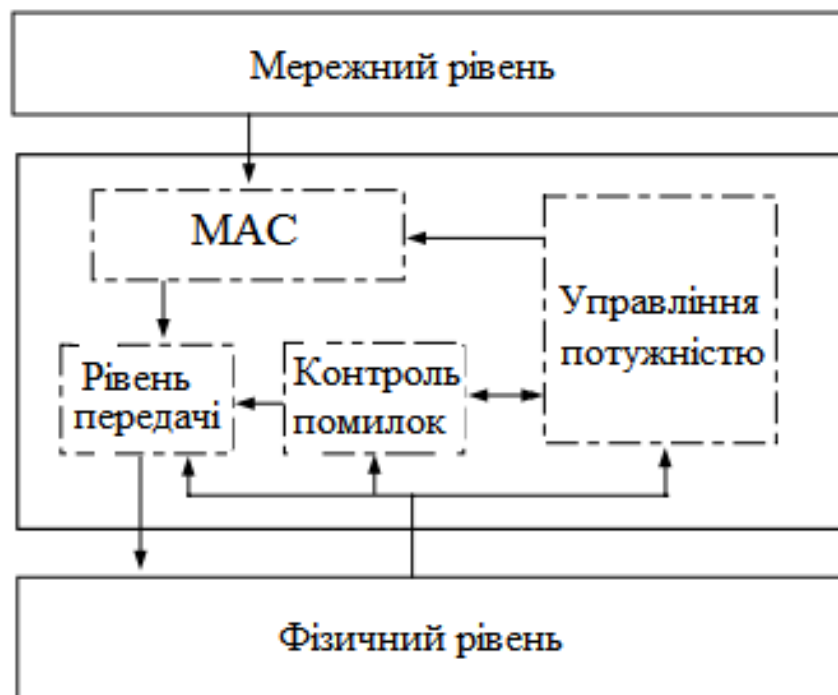


Рисунок 1.15 – Управління потужністю в БСМ

Отже, недоліком механізму управління потужністю трансивера є зростання накладних витрат: обчислювальних ресурсів, пам'яті та енергії.

1.4.2 Автоматичний запит повторної передачі

Схема контролю помилок на основі автоматичного запиту повторної передачі (ARQ) здійснює повторну передачу втрачених або пошкоджених

пакетів. Для реалізації даної схеми до пакету додається додатковий заголовок, в якому вказується вихідний вузол, і алгоритм виявлення помилок з низькою обчислювальною складністю. Як правило, контроль цілісності пакетів даних у БСМ, фізичний рівень яких відповідає стандарту IEEE 802.15.4, здійснюється на основі циклічного надлишкового коду (Cyclic Redundancy Check, CRC). Перевірочна послідовність (Frame check sequence, FCS) має довжину 16 біт і розраховується по MAC header (MHR) і MAC payload частинах кадру (рисунок 1.16). Розрахунок FCS проводиться на основі стандартного полінома генератора степені 16 [156]:

$$G_{16}(x) = x^{16} + x^{12} + x^5 + 1.$$

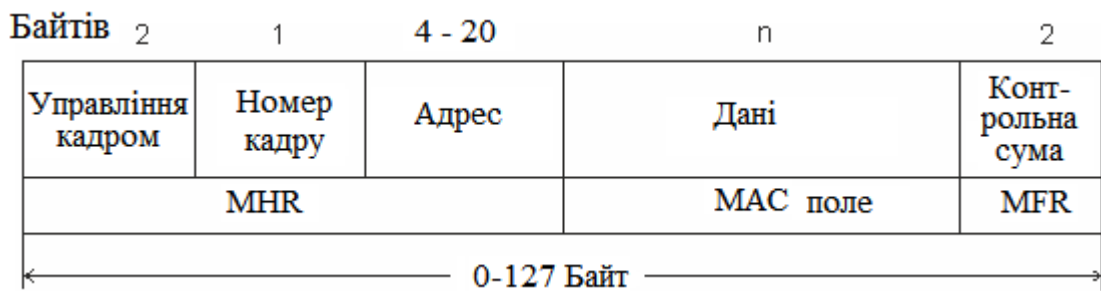


Рисунок 1.16 – Структура фрейму даних стандарту IEEE 802.15.4

Використання CRC коду дозволяє тільки виявити наявність помилки в зазначених блоках даних. При виявленні помилки надсилається запит на повторну передачу втраченого або спотвореного пакету. Контрольна сума додається до кінця пакета. Вузол приймача обчислює контрольну суму прийнятих бітів і порівнює його з контрольною сумою отриманого пакета.

Приймач повідомляє джерело даних про цілісність прийнятого пакета. Це може бути виконано шляхом надсилання позитивного підтвердження АСК для кожного успішно прийнятого пакету або негативного підтвердження NACK для кожного пошкодженого пакету. У випадку використання пакетів АСК, відправник встановлює таймер для виявлення чи

надійшли вказані пакети. Якщо відправник робить висновок, що пакет пошкоджений, відбувається повторна передача.

В схемах ARQ використовують наступні механізми: “повторити передачу N – пакетів”, “вибірковий повтор”, “зупинити передачу і очікувати” [133, 168, 192].

При використанні схеми “Повторити передачу N – пакетів”, здійснюється передача N – пакетів і, якщо не прийшов пакет підтвердження, (ACK) всі пакети передаються повторно. При вибіркового повторі відправник передає тільки ті пакети, на які не отримали пакету підтвердження. Схема “зупинити передачу і очікувати” зупиняє передачу і очікує пакету підтвердження, і тільки після його отримання відправляє наступний пакет.

В [186] показано, що схема ARQ є енергетично неефективною оскільки споживає більше енергії у зв'язку з повторною передачею кожного помилкового пакету. Дослідження показують, що затрати енергії на передачу одного біту інформації приблизно дорівнюють затратам на виконання тисячі операцій в типовому процесорі сенсорного вузла [119].

Недоліком використання механізмів контролю помилок ARQ в БСМ є зниження корисної пропускну здатності мережі та додаткові витрати на повторну передачу пакетів.

1.4.3 Пряме виправлення помилок (FEC)

Пряме виправлення помилок використовує коректуючі коди, які забезпечують виправлення помилок в пакеті даних за рахунок додавання контрольних символів (рисунок 1.17) [8, 159, 209].

В коректуючому коді (n, k, t) , де n – загальна довжина коду, k – кількість інформаційних бітів, можуть бути виправлені до t бітів без використання повторної передачі пакетів. Однак використання коректуючих кодів вносить додаткове навантаження на мережу (зменшує корисну

пропускну здатність мережі, а також зростають витрати енергії на декодування пакетів), яке пов'язано з передачею та прийманням надлишкових бітів та необхідністю декодування пакетів.

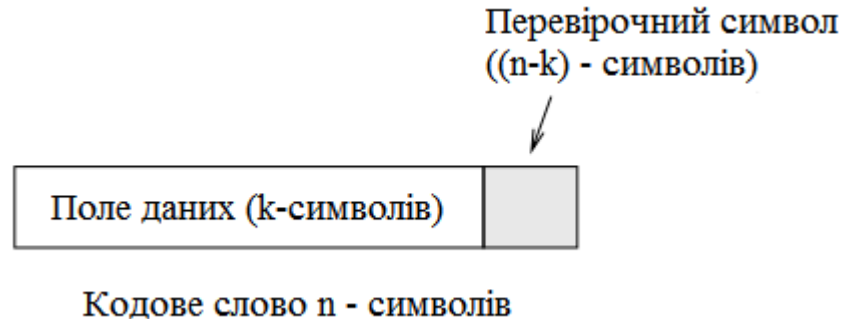


Рисунок 1.17 – Пряме виправлення помилок в БСМ

Отже, при виборі коректуючих кодів необхідно враховувати швидкість коду, яка визначає надлишковість та складність алгоритмів кодування/декодування.

В схемах прямого виправлення помилок використовують лінійні блокові коди (БЧХ, Ріда-Соломона), згорткові коди, турбо коди та інші [139, 151].

В [139] показано, що при ймовірності помилок $BER > 10^{-4}$ використання повторної передачі пакетів забезпечує більш ефективне споживання енергії порівняно з використанням коректуючих кодів. Це пов'язано з тим, що при малій ймовірності помилок (BER) повторна передача пакетів відбувається рідко, а при використанні коректуючих кодів надлишковість присутня в кожному пакеті.

Використання коректуючих кодів, як показано в [151], дозволяє також при заданій потужності (- 5 дБм) збільшити ефективну дальність передачі на 45% в порівнянні з автоматичним повторним запитом передачі.

1.4.4 Гібридна схема повторної передачі пакетів (HARQ)

Протоколи з ARQ, як правило, ефективні, коли ймовірність помилки низька, оскільки надлишкові біти не передаються. Однак ці протоколи

стають неефективними, коли ймовірність помилки зростає, оскільки при спотворенні хоча б одного біту необхідно повторно передавати весь пакет. З іншого боку, використання коректуючих кодів стає неефективним, коли ймовірність помилки низька, оскільки надлишкові біти відправляються з пакетом але не використовуються.

Гібридна схема повторної передачі пакетів (HARQ) використовує переваги обох схем контролю помилок FEC і ARQ. При надходженні запиту на повторну передачу пакетів схема HARQ поступового підвищує стійкість пакету до помилок через використання більш потужних коректуючих кодів.

Розрізняють два типи гібридних схем: HARQ-I і HARQ-II. В схемі HARQ-I відправляється некодований пакет, і у разі помилки передавач здійснює повторну передачу пакету з використанням коректуючих кодів. В схемі HARQ-II при запиті на повторну передачу відправляються тільки коректуючі символи. Це забезпечує ефективне використання пропускну здатності мережі та економію енергії.

Схема HARQ має переваги для окремих областей застосування БСМ [132, 133], але вона споживає значну кількість енергії [162, 168].

1.4.5 Вибір коректуючих кодів в схемах прямого виправлення помилок

Вибір оптимального коду для виправлення помилок в БСМ широко вивчається багатьма дослідниками в останнє десятиліття [25, 35, 109, 206, 208], але стандартом тип коректуючих кодів до цього часу не встановлено.

Є два основних типи кодів корекції помилок: блокові коди і згорткові коди. Серед поширених блокових кодів слід зазначити наступні: коди Хемінга [25], Голя (Golay) [37], Боуза - Чоудхурі - Хоквінгема (БЧХ) [37], Ріда-Соломона [45] та код низької щільності контролю парності (LDPC - код) [35], який також відносять до класу блокових кодів.

Результати моделювання показують, що застосування кодів БЧХ може покращити ефективність використання енергії до 23% порівняно з

некодованою передачею даних та на 15% порівняно з використанням згорткових кодів.

В [123] досліджено коди БЧХ, Ріда-Соломона і згортковий код з позиції складності та енергоспоживання при реалізації на ПЛІС і спеціалізованих інтегральних схемах (ASIC). Результати показують, що згорткові коди мають переваги порівняно з кодами БЧХ і Ріда-Соломона, але в той же час декодер Вітербі споживає більше енергії, ніж передача некодованих даних, тому згорткові коди недоцільно використовувати в БСМ з автономним живленням вузлів. При цьому зазначено, що двійковий код БЧХ з реалізацією на ASIC є кращим вибором для БСМ, які працюють на відкритому просторі.

В [109] отримано результати, які показують, що ефективність використання енергії механізму ARQ не залежить від ретрансляції пакетів, але залежить від розміру пакета і відстані між вузлами.

В роботі [161] автори пропонують використовувати коди контролю парності з низькою щільністю (LDPC) і для каналного кодування, і для кодування джерела з метою зниження енергоспоживання передачі в БСМ. Отримані результати моделювання показують, що при використанні кодів LDPC затрати енергії зростають на 42% порівняно з кодами БЧХ.

В [120] запропоновано адаптивну схему, в якій вихідний вузол має можливість змінювати схему корекції помилок для кожної передачі. В роботі досліджено енергетичну ефективність адаптивної та неадаптивної схем корекції помилок на різних відстанях зв'язку та розмірах пакетів. Результати показують, що адаптивна схема має більшу енергоефективність порівняно з неадаптивною на всіх відстанях і розмірах пакетів.

В роботі [134] використано коди Ріда-Соломона (RS) для кодування в вихідному вузлі і декодування тільки на базовій станції, щоб зберегти енергію на рівні вузла, і, тим самим, продовжити тривалість роботи мережі.

В [126, 198] дослідження затрат енергії для різних схем модуляції та різної довжини кодів Ріда-Соломона показали, що код Ріда-Соломона з

параметрами (63, 59, 5) є оптимальним порівняно з передачею без використання кодування. В роботі [161] запропоновано схему, в якій для кодування даних в передавальному вузлі використовують LDPC-код і часткове декодування виконується в проміжних вузлах, а повне декодування виконується тільки на базовій станції.

Оцінка продуктивності використання різних кодів показує, що більш потужні коди забезпечують кращу продуктивність, але є енергетично неефективними, в порівнянні з простими кодами, які мають низьку продуктивність, але є більш енергоефективним.

При виборі схеми контролю помилок в БСМ необхідно також враховувати стратегію контролю помилок (“вузол – вузол”, “з кінця в кінець”).

Потужні коди є оптимальними для використання управління помилками “з кінця в кінець” в той час, як прості коди є кращими для стратегії контролю помилок в “вузол – вузол”.

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

До коректуючих кодів в БСМ висуваються наступні вимоги:

- 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 [119, 151]. Однак використання кодів Ріда-Соломона унеможлиблює використання адаптивних схем контролю помилок. Оскільки при збільшенні /зменшенні кількості перевірочних символів необхідно перераховувати всі перевірочні символи

[25]. Для вибору ефективної схеми контролю помилок та типу коректуючого коду необхідно також враховувати допустиму ймовірність помилки для заданої БСМ та характеристики каналу зв'язку.

1.4.6 Обробка даних в системі залишкових класів

Система залишкових класів (СЗК) належить до непозиційних систем числення і базується на понятті залишку та китайській теоремі про залишки [2, 75, 182]. Будь-яке ціле додатне число в СЗК подається у вигляді набору найменших додатних залишків від ділення цього числа на вибрану систему взаємно простих чисел (модулів) p_1, p_2, \dots, p_n . При цьому, діапазон подання чисел у вибраній системі модулів $0 \leq X < \wp$, де $\wp = \prod_{i=1}^n p_i$.

Позначимо x_i найменший додатній залишок від ділення X на p_i [2]:

$$x_i = X - \left[\frac{X}{p_i} \right] \cdot p_i,$$

або

$$x_i = X \pmod{p_i} = |X|_{p_i}, \quad (1.1)$$

де $[\bullet]$ – заокруглення до меншого цілого,

n – кількість модулів.

Тоді число X в СЗК запишемо у вигляді [2, 182]:

$$X = \{x_1, x_2, \dots, x_i, \dots, x_n\}.$$

Зворотне перетворення здійснюється за формулою [2]

$$X = \left(\sum_{i=1}^n x_i \cdot M_i \right) \pmod{\wp}, \quad (1.2)$$

де $M_i = \delta_i \cdot m_i$, $\delta_i = \frac{\wp}{p_i}$, $m_i = \delta_i^{-1} \pmod{p_i}$.

В [39, 40, 42, 43] розроблено та досліджено досконалу форму системи залишкових класів, при якій система модулів p_i вибирається таким чином, що всі значення коефіцієнтів $m_i = 1$. Однак досконала форма СЗК має обмежене використання, оскільки в результаті перетворення отримуємо залишки різної розрядності, неможливо підібрати системи модулів малої (5-8 біт) розрядності.

Арифметичні операції над числами в СЗК. Виконаємо арифметичні операції над числами X , Y , поданими в СЗК: $X = (x_1, x_2, \dots, x_n)$, $Y = (y_1, y_2, \dots, y_n)$.

Додавання:

$$X + Y = (x_1, x_2, \dots, x_n) + (y_1, y_2, \dots, y_n) = (z_1, z_2, \dots, z_n),$$

$$\text{де } z_i = |x_i + y_i|_{p_i},$$

віднімання:

$$X - Y = (x_1, x_2, \dots, x_n) - (y_1, y_2, \dots, y_n) = (z_1, z_2, \dots, z_n),$$

$$\text{де } z_i = |x_i - y_i|_{p_i},$$

множення:

$$X \times Y = (x_1, x_2, \dots, x_n) \times (y_1, y_2, \dots, y_n) = (z_1, z_2, \dots, z_n),$$

$$\text{де } z_i = |x_i \times y_i|_{p_i}.$$

Перевагою виконання арифметичних операцій в СЗК є відсутність переносів між розрядами.

Обробка та передача даних в системі залишкових класів має ряд переваг завдяки незалежності, малорозрядності та рівноправності залишків та можливості паралельного виконання арифметичних операцій. Однак в даний час СЗК використовується тільки при вирішенні деяких спеціалізованих задач, що обумовлено необхідністю перетворення двійкового коду, в якому

працюють універсальні комп'ютери та пристрої обробки даних, в код СЗК та зворотного перетворення при поданні інформації користувачу [7,179, 180, 181, 1907].

Крім вище названих переваг СЗК для обробки даних, також розроблені ефективні коректуючі коди СЗК здатні виявляти та виправляти пакети помилок [2, 3, 29, 30, 75, 76, 146, 182, 189, 205].

Нехай в системі залишкових класів з модулями $p_1, p_2, \dots, p_i, \dots, p_k$, число X подається залишками

$$X = (x_1, x_2, \dots, x_i, \dots, x_k),$$

де X – код повідомлення, $x_i = X \pmod{p_i}$, k – кількість інформаційних модулів.

Відповідно, у коректуючих кодах СЗК, значення контрольного розряду визначається [2, 75]:

$$x_{k+1} = X \pmod{p_{k+1}},$$

де p_{k+1} – перевірочний модуль, $p_k < p_{k+1}$.

З вибраного набору модулів перші k модулів використовуються для подання даних, а $r = n - k$ модулів використовуються для виявлення та виправлення помилок. Набір із k модулів утворює робочий діапазон:

$$P_w = \prod_{i=1}^k p_i,$$

при цьому загальний діапазон визначається як $\wp = P_w \cdot \prod_{i=1}^{n-k} p_i = \prod_{i=1}^n p_i$.

Отже, повідомлення X подаються за допомогою розширеної системи модулів $X = (x_1, x_2, \dots, x_i, \dots, x_k, x_{k+1}, \dots, x_{k+r})$, при цьому обчислення контрольного розряду потребує звернення до початкового значення числа X , а це вимагає додаткових обчислювальних ресурсів.

Введення одного контрольного модуля дозволяє виявляти помилки в залишку за одним модулем. Для виправлення помилки в одному символі (залишку) необхідно ввести два контрольні модулі. Алгоритм виявлення та виправлення помилок базується на обчисленні проекції числа. Проекцією числа A за модулями p_i, p_j називається число A_{ij} , отримане із A закресленням цифр за модулями p_i, p_j [2]. Отже, якщо в системі з модулями p_1, p_2, \dots, p_k , модулі p_{k+1}, p_{k+2} є контрольними і, якщо число

$$A = (\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_k, \alpha_{k+1}, \alpha_{k+2})$$

є правильним, то всі проекції числа A за модулями p_i і p_j рівні між собою і співпадають з величиною A , тобто

$$A_{ij} = A < P,$$

(при $i, j = 1, 2, \dots, k + 2, i \neq j$) та за умови, що всі проекції A_{ij} числа A рівні між собою і співпадають за величиною з числом A , то число A правильне. Відповідно, якщо в системі $p_1, p_2, \dots, p_k, p_{k+1}, p_{k+2}$ з двома контрольними модулями задано неправильне число $\tilde{A} = (\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_i, \dots, \tilde{\alpha}_k, \tilde{\alpha}_{k+1}, \tilde{\alpha}_{k+2})$, то необхідною і достатньою умовою, що цифра $\tilde{\alpha}$ з помилкою в \tilde{A} , є правильна її проекція \tilde{A}_i за модулем p_i .

Отже, для виявлення помилки необхідно обчислити проекції числа \tilde{A} за всіма модулями. Враховуючи, що для обчислення проекції необхідне відновлення позиційного подання числа \tilde{A} , даний алгоритм потребує значних обчислювальних ресурсів.

Оскільки виявлення та виправлення помилок відбувається з використанням алгоритмів зворотного перетворення, тобто потребує відновлення позиційного подання повідомлення, то це значно ускладнює

процес декодування при зростанні кількості символів у повідомленні та розрядності символів [2].

В [146] розроблено алгоритм виявлення помилок, в якому відновлення позиційного подання повідомлення відбувається окремо за модулями робочого і перевірного діапазонів:

$$Y = \left(\sum_{i=1}^n y_i \cdot M_i \right) \bmod P_w,$$

$$E = \left(\sum_{i=1}^n y_i \cdot M_i \right) \bmod P_c,$$

де y_i – символи прийнятого повідомлення, $P_c = \prod_{i=k+1}^n p_i$,

та обчисленням синдрому:

$$s = (Y - E) \bmod P_c.$$

Виправлення помилки здійснюється за таблицею синдромів, в якій кожному значенню помилки відповідає одне значення синдрому. Даний підхід дозволяє зменшити кількість необхідних обчислень, однак залишається проблема відновлення позиційного подання чисел при зростанні кількості символів у повідомленні. Наприклад, для системи модулів $p_1 = 5$, $p_2 = 13$, $p_3 = 17$, $p_4 = 19$, $p_5 = 23$, $p_6 = 29$. Базисні числа мають вигляд: $M_1 = 5601466$, $M_2 = 12926460$, $M_3 = 9884940$, $M_4 = 11792560$, $M_5 = 13394810$, $M_6 = 2414425$. Відповідно, при зростанні значень та кількості модулів значно збільшується значення базисних чисел, що, в свою чергу, призводить до зростання апаратних та обчислювальних затрат.

Коректуючі коди СЗК також мають обмеження, які накладаються умовою вибору зростаючої послідовності взаємно простих модулів. При цьому необхідно, щоб перевірочні модулі були однакової розрядності. В

іншому випадку виникає проблема ефективного (оптимального) збереження перевірочних символів. Для ефективного використання даних кодів в безпроводних сенсорних мережах необхідно попередньо конвертувати дані в СЗК, що потребує додаткових затрат енергії та обчислювальних ресурсів.

В [3] розроблені слабо арифметичні коди СЗК, які мають обмеження при контролі виконання арифметичних операцій, однак можуть ефективно використовуватись для виявлення та виправлення помилок в процесі передачі даних, оскільки дозволяють простіше формувати контрольні символи.

В слабо арифметичних кодах СЗК значення контрольного розряду визначається [3]:

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

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

Припустимо, що в процесі передачі відбулась помилка в одному із залишків і замість A отримали A' :

$$A' = (b_1, b_2, \dots, b'_i, \dots, b_k, \bar{b}_{k+1}).$$

Значення контрольного розряду обчислюємо за формулою

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

Виявлення помилки відбувається за рахунок обчислення різниці між розрахованим і прийнятим контрольними символами $\delta = \bar{b}'_{k+1} - \bar{b}_{k+1}$. Отже, при $\delta = 0$ – помилка відсутня; при $\delta \neq 0$ – є помилка.

Для виявлення помилки в будь-якому з блоків необхідно, щоб кожному значенню δ_i відповідало одне значення помилки e_i .

Недоліком слабо арифметичних кодів СЗК є те, що дані, які необхідно обробляти або передавати, мають бути подані в СЗК, що обмежує їх використання, оскільки вимагає додаткового перетворення з двійкової системи числення в СЗК і проведення зворотного перетворення.

Враховуючи викладене вище, виникає потреба в розробці коректуючих кодів, які б зберігали основні переваги кодів СЗК, але обробляли повідомлення подані в позиційних системах числення.

1.5 Аналіз критеріїв оцінки ефективності безпроводних сенсорних мереж

Під ефективністю розуміють властивість системи виконувати поставлені задачі в заданих умовах використання і з певною якістю [18, 36, 74, 23]. Поняття ефективності пов'язано з отриманням деякого корисного результату – ефекту використання мережі. Ефект досягається ціною витрат певних ресурсів, тому ефективність мережі часто розглядається у вигляді співвідношення між ефектом (виграшем) і витратами [36].

Показники ефективності характеризують ступінь пристосованості системи до виконання поставлених перед нею завдань і є узагальнюючими показниками оптимальності функціонування системи. До факторів, які визначають ефективність функціонування системи відносяться: якість, умови функціонування та способи використання [36]

При цьому показники ефективності, як правило, поділяють на три групи [206]:

$$E = \{E_a, E_t, E_e\},$$

де E_a – показники ефективності використання (цільового застосування) системи;

E_t – показники технічної ефективності системи;

E_e – показники економічної ефективності системи.

При створенні БСМ та оцінці їх ефективності, крім технічних, необхідно також враховувати економічні показники. Одним з таких підходів є принцип мінімальних витрат, згідно з яким найкращою вважається та система, для реалізації і експлуатації якої потрібні найменші витрати при заданому технічному ефекті [36].

В даний час використовуються різні підходи до вибору критеріїв оцінки ефективності функціонування БСМ. При цьому відсутні методики, які дозволяють проводити комплексну оцінку ефективності роботи БСМ. Як правило, для оцінки ефективності роботи БСМ найчастіше використовують критерії за одним або декількома параметрами. Відповідно, і порівняння ефективності роботи БСМ здійснюється за окремо визначеними критеріями в залежності від функцій та цільового призначення БСМ.

В [18, 7373] виділено три критерії оцінки ефективності безпроводних сенсорних мереж, орієнтованих на вирішення типових задач: критерій сумарної середньої затримки передачі, максимальної затримки та середньої затримки передачі. Наведені критерії враховують тільки часові характеристики роботи мережі і є важливими для оцінки БСМ, які працюють в системах реального часу.

В [73] для оцінки ефективності мережі пропонується узагальнений показник ефективності, який враховує продуктивність, швидкість та точність передачі даних.

В [71] використано вагові критерії при оцінці ефективності БСМ. При цьому значення інтегрального критерію ефективності визначається:

$$k = \sum_{i=1}^n (\alpha_i \cdot k_i),$$

де n – кількість критеріїв,

k_i – значення критерію,

α_i – вагові коефіцієнти.

Для усунення впливу одного критерію на інший їх значення використовуються в приведеній до максимуму формі. Однак, враховуючи значу кількість параметрів при оцінці роботи БСМ, правильний вибір вагових коефіцієнтів є складною задачею.

При виборі та порівнянні безпроводних сенсорних мереж необхідно оцінювати параметри, які дозволять системі найбільш ефективно виконувати поставлені перед нею завдання.

Серед показників, які доцільно враховувати при оцінці ефективності БСМ, необхідно виділити наступні [68, 23]:

1) коефіцієнт використання каналу за потужністю (енергетична ефективність):

$$\beta = \frac{R}{\rho_0},$$

де R – швидкість передачі даних;

ρ_0 – відношення потужності сигналу P_c до спектральної густини N_0 потужності завад;

2) коефіцієнт використання каналу за смугою частот (частотна ефективність):

$$\gamma = \frac{R}{F},$$

де F – ширина смуги частот, яку займає сигнал;

3) коефіцієнт використання каналу за пропускною здатністю (інформаційна ефективність):

$$\eta = \frac{R}{F},$$

який є узагальненою характеристикою системи передачі даних;

4) коефіцієнт ефективності кодування:

$$\lambda = \frac{H(x)}{l_s},$$

де $H(x)$ – ентропія повідомлення,

$$l_s - \text{середня довжина кодової комбінації}, \quad l_s = \sum_{i=1}^M P_i \cdot l_i,$$

M – кількість кодових комбінацій,

P_i – ймовірність появи символу,

l_i – кількість бітів в i -й кодовій комбінації;

5) коефіцієнт, який враховує надлишковість коректуючих кодів (ефективність коректуючих кодів):

$$r = \frac{k}{n},$$

де k – кількість інформаційних символів,

n – загальна довжина коду;

6) коефіцієнт, який враховує накладні витрати протоколів передачі даних (ефективність протоколу передачі даних):

$$\mu = \frac{d}{d + d_s},$$

де d – довжина поля даних протоколу передачі,

d_s – довжина службових даних в структурі протоколу передачі.

В безпроводних системах передачі даних, до яких належать безпроводні сенсорні мережі, в якості критерію ефективності використовують відношення E_b/N_0 , де E_b – енергія біту, N_0 – спектральна густина потужності шуму.

Ефективність кодування G визначається, як зменшення E_b/N_0 , яке досягається при використанні завадостійкого кодування [60]:

$$G = \left(\frac{E_b}{N_0} \right)_u - \left(\frac{E_b}{N_0} \right)_c \text{ (dB)}, \quad (1.3)$$

де $(E_b/N_0)_u$ і $(E_b/N_0)_c$ – необхідне значення E_b/N_0 без кодування і з кодуванням.

Враховуючи значну кількість параметрів, які впливають на ефективність роботи БСМ, необхідно формувати критерії ефективності на етапі проектування в залежності від вимог до якості, умов функціонування та способів використання сенсорних мереж.

Аналіз літератури показав, що при оцінці ефективності БСМ недостатньо уваги приділяється вибору методів обробки даних на різних рівнях сенсорних мереж, які необхідно враховувати на етапі проектування БСМ, зокрема ефективності кодування даних, ефективності використання коректуючих кодів та протоколів передачі даних.

Оскільки БСМ використовують відкрите середовище передачі радіосигналів, яке доступне практично будь-якому користувачеві в зоні дії антени, то питання забезпечення високої надійності функціонування таких мереж є ключовим завданням.

Враховуючи, що проблема підвищення ефективності функціонування БСМ тісно пов'язана з проблемою забезпечення високої надійності передачі даних, відповідно, в роботі в якості основних критерії для оцінки ефективності запропонованих рішень вибрано надійність та продуктивність мережі. Продуктивність мережі в свою чергу включає такі параметри, як швидкість передачі даних, пропускну здатність та затримку передачі [5].

1.6 Напрями підвищення ефективності роботи безпроводних сенсорних мереж

Враховуючи наведені вище критерії, цільову функцію проектування БСМ можна записати у вигляді [103]:

$$F_{WSN} = \{C_{\max}, Z_{\max}, E_{\min}, QoS, \theta_{\min}, O_{\min}, \Sigma_{\min}\},$$

де C_{\max} – корисна пропускна здатність;

Z_{\max} – надійність передачі даних;

E_{\min} – затрати енергії на перетворення, обробку та передачу даних;

QoS – якість обслуговування (смуга пропускання; затримка передачі пакету; коливання затримки при передачі пакетів; втрата пакетів (визначає кількість втрачених пакетів в мережі під час передачі));

θ_{\min} – часова складність;

O_{\min} – апаратна складність;

Σ_{\min} – затрати на реалізацію БСМ.

Наведена цільова функція дозволяє створити оптимальну БСМ, однак в залежності від застосування необхідно виділити основні і другорядні критерії. В роботі в якості основних критеріїв вибрано корисну пропускну здатність та надійність передачі даних.

Підвищити ефективність роботи БСМ пропонується за рахунок використання наступних підходів:

1. Максимальна обробка вхідних даних у безпроводних вузлах (розробка методів стиснення, агрегування даних та кодування даних).
2. Підвищення корисної пропускну здатності мережі (зменшення кількості службових даних в протоколах передачі, збільшення розміру поля даних).

3. Підвищення загальної пропускної здатності каналів зв'язку (використання методів мережного кодування даних; протоколів багатошляхової маршрутизації).

4. Підвищення надійності передачі даних (використання коректуючих кодів та адаптивних схем контролю помилок).

5. Розробки енергоефективних методів і алгоритмів перетворення, обробки та передачі та даних (використання переваг системи залишкових класів та модулярної арифметики).

6. Мінімізація обчислювальних та апаратних затрат на всіх етапах руху даних.

Запропоновані підходи забезпечать підвищення загальної пропускної здатності мережі, надійності передачі даних та тривалості роботи від автономного живлення, що в свою чергу дозволить розширити область ефективного застосування безпроводних сенсорних мереж, зокрема при зборі та обробці мультимедійних даних.

Висновки до першого розділу

1. Проведено аналіз стану розробки безпроводних сенсорних мереж та областей їх застосування. Розроблено узагальнену класифікацію безпроводних сенсорних мереж за наступними ознаками: технологією та середовищем передачі, типом вузла, типом живлення, площею розгортання та областю застосування.

2. Розкрито переваги використання програмованих логічних інтегральних схем при проектуванні вузлів безпроводних сенсорних мереж, зокрема, в якості апаратних платформ безпроводних мультимедійних сенсорних мереж. Застосування ПЛІС робить процес розробки більш гнучким, оскільки дозволяє використовувати готові програмні ядра мікропроцесорів з необхідним набором периферійних контролерів,

забезпечує високу обчислювальну потужність і можливість реконфігурації вузлів.

3. Проведено аналіз методів підвищення корисної пропускної здатності безпроводних сенсорних мереж. Серед найбільш перспективних виділено методи мережного кодування, методи стиснення та агрегування даних.

4. Проведено аналіз методів підвищення надійності передачі даних в безпроводних сенсорних мережах. Виділено переваги та недоліки схем контролю помилок для використання в безпроводних сенсорних мережах. Показано переваги використання системи залишкових класів для обробки та передачі даних в БСМ.

5. Проаналізовано підходи до оцінки ефективності безпроводних сенсорних мереж. Сформовано критерії, які доцільно враховувати на етапі проектування та при оцінці ефективності БСМ.

6. Запропоновано підходи до підвищення ефективності безпроводних сенсорних мереж, що, в свою чергу, дозволить розширити область застосування даного типу мереж, зокрема в системах автоматизації та при обробці мультимедійних даних в системах відеоспостереження побудованих на основі БСМ.

РОЗДІЛ 2

МЕТОДИ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖ НА ОСНОВІ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ

2.1 Метод мережного кодування даних на основі системи залишкових класів

Методи мережного кодування почали активно розвиватися з роботи Ahlswede R. [111]. Основною проблемою мережного кодування є вибір методу комбінування пакетів. На даний час мережне кодування розвивається в двох теоретичних напрямках: на основі китайської теореми про залишки та лінійне кодування [129, 142, 221, 230]. Існуючі методи лінійного мережного кодування призводять до значного збільшення службових даних в пакеті протоколу, тому не можуть бути використані в БСМ, які характеризуються малими розмірами пакетів і мають обмежену пропускну здатність каналів зв'язку [111, 142, 230].

На відміну від методів лінійного мережного кодування, мережне кодування на основі китайської теореми про залишки дозволяє зменшити обчислювальні витрати на кодування даних та обсяги службових даних в протоколах передачі [129, 221, 230]. Тому автором запропоновано новий метод мережного кодування.

Ідею запропонованого методу мережного кодування розглянемо на прикладі топології мережі (рисунок 2.1). Нехай потрібно передати повідомлення X_1 і X_2 вузлу S [95, 84, 222, 223].

Вибираємо взаємно прості модулі p_1 , p_2 , p_3 , p_4 . У вузлі A повідомлення X_1 ділимо на модулі p_1 , p_2 , отримаємо $b_1 = X_1(\text{mod } p_1)$, $b_2 = X_1(\text{mod } p_2)$.

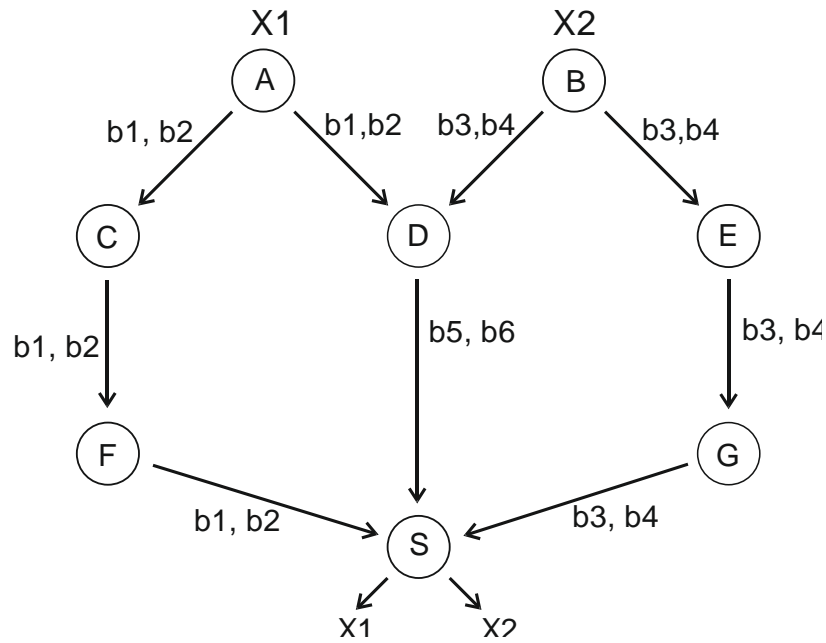


Рисунок 2.1 – Передача залишків при мережному кодуванні
(схема 1)

У вузлі B повідомлення $X2$ ділимо на модулі $p3$, $p4$ і отримуємо $b3 = X2(\text{mod } p3)$, $b4 = X2(\text{mod } p4)$. Отримані залишки передаємо відповідними маршрутами (див. рисунок 2.1).

У проміжному вузлі D відбувається об'єднання залишків $b1, b2, b3, b4$ за формулою [2]:

$$X = \left(\sum_i^n B_i \cdot b_i \right) \text{mod } R, \quad (2.1)$$

$$\text{де } R = \prod_{i=1}^n p_i,$$

n – кількість модулів,

$$B_i \text{ – базисні числа, } B_i = \delta_i \cdot m_i, \delta_i = \frac{R}{p_i}, \delta_i \cdot m_i \equiv 1(\text{mod } p_i),$$

m_i – коефіцієнт, який знаходиться в межах $0 < m_i < p_i$,

і формування нових залишків за модулями $p2$, $p3$:

$$b5 = X \pmod{p2},$$

$$b6 = X \pmod{p3},$$

або

$$b5 = X \pmod{p5},$$

де $p5 = p2 \cdot p3$.

Для передачі повідомлень $X1$, $X2$ в мережі (див. рисунок 2.1) необхідно передати сім пакетів даних (рисунок 2.2).

Пакет даних складається із залишків та значень взаємно простих модулів і має вигляд:

$$\{b_1, b_2, \dots, b_i \dots, b_n \mid p_i, p_{i+1}\}.$$

В вузлі S відбувається об'єднання залишків за формулою (2.1) та обчислення значень $X1 = X \pmod{p1 \cdot p2}$ і $X2 = X \pmod{p3 \cdot p4}$.

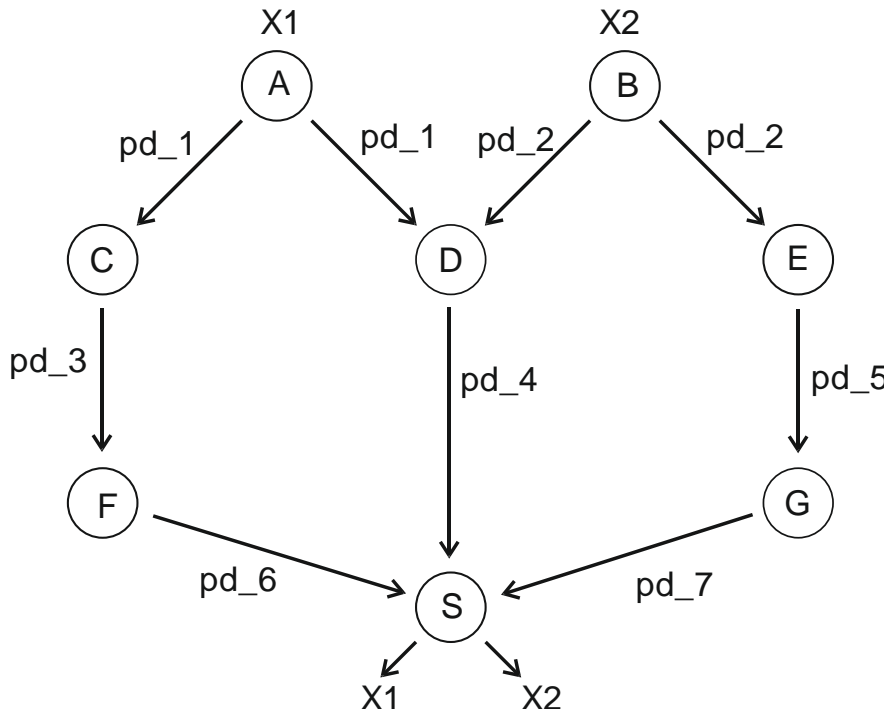


Рисунок 2.2 – Розподіл пакетів при мережному кодуванні для схеми 1

Дана схема кодування має переваги при багатоадресній передачі, однак є надлишковою для наведеної топології, тобто при передачі “багато до одного”.

З врахуванням наведеного вище, запропоновано новий метод мережного кодування, при якому отримані залишки b_1 , b_2 у вузлі A передаються різними маршрутами [95, 223]. Наприклад, залишок b_1 передається маршрутом $A \rightarrow C$, а b_2 передається маршрутом $A \rightarrow D$. Аналогічно розподіл залишків відбувається у вузлі B (рисунок 2.3).

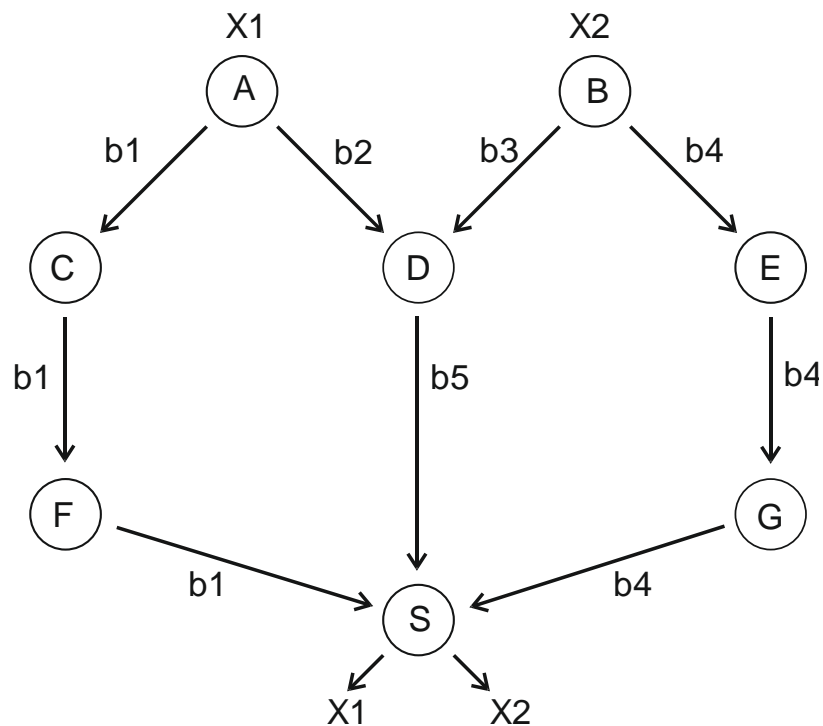


Рисунок 2.3 – Передача залишків при мережному кодуванні
(схема 2)

При даному розподілі залишків для передачі повідомлень X_1 , X_2 у вузол S необхідно передати дев'ять пакетів (рисунок 2.4).

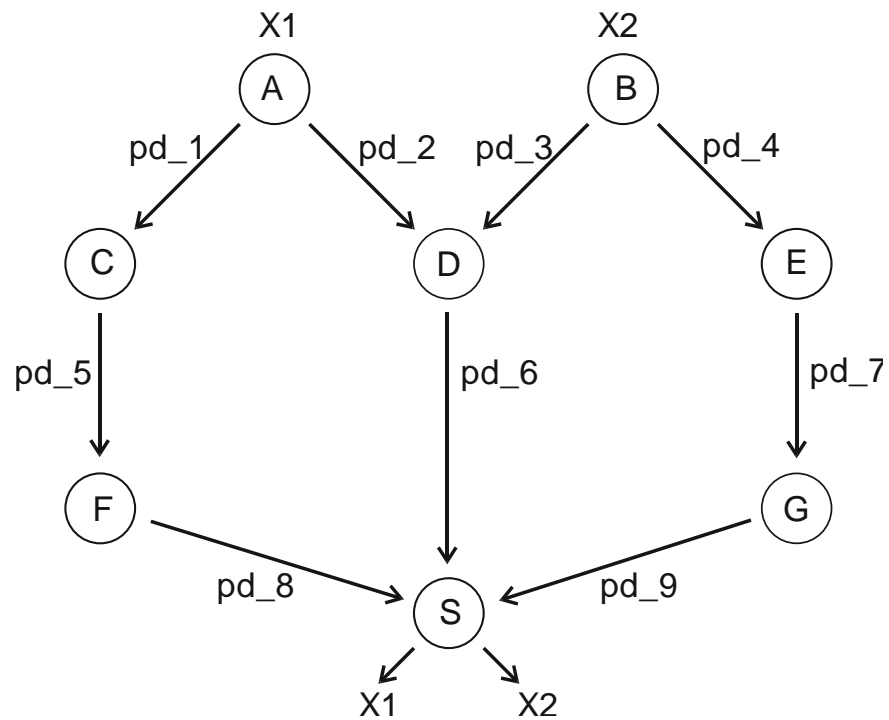


Рисунок 2.4 – Розподіл пакетів при мережному кодуванні для схеми 2:
pd_i – номер пакету даних

Незважаючи на те, що запропонований метод потребує передачі більшої кількості пакетів, загальний обсяг даних, які необхідно передати, є меншим за рахунок передачі залишків різними маршрутами. Ще однією перевагою запропонованого методу є можливість рівномірного розподілу навантаження в мережі. Для цього взаємно прості модулі вибираємо таким чином, щоб по незалежних (окремих) маршрутах (A-C-F-S) та (B-E-G-S) (див. рисунок 2.4) передавались залишки більшого обсягу, а по спільному маршруту (D-S) – залишки меншого обсягу.

Для зменшення кількості пакетів аналогічно схемі (див. рисунок 2.2) необхідно передати через перший вузол одночасно залишки за двома модулями (рисунок 2.5), а на проміжних вузлах ретранслювати тільки оригінальні залишки (які не передавались іншими маршрутами).

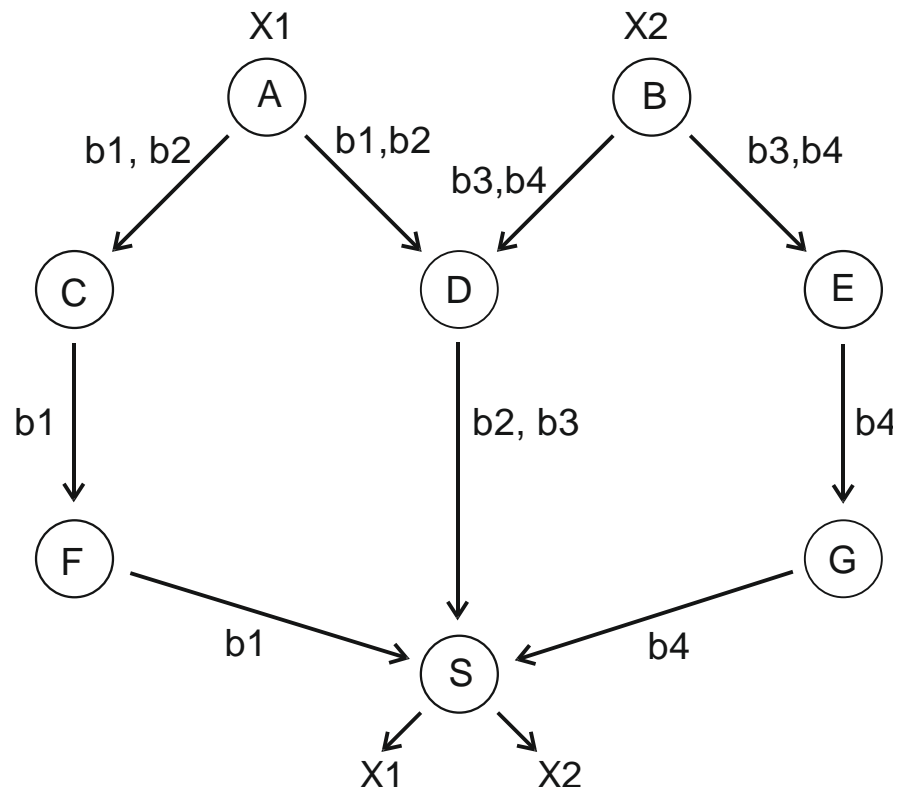


Рисунок 2.5 – Передача залишків при мережному кодуванні (схема 3)

2.1.1 Теоретичні дослідження методу мережного кодування

Розрахунок обсягу даних при мережному кодуванні проведемо на прикладі кодування повідомлень довжиною 8 біт, тобто $X \in [0 \dots 255]$. Для передачі залишків виберемо схему 3. Вибираємо взаємно прості модулі $p_1 = 27$, $p_2 = 11$, $p_3 = 13$, $p_4 = 23$. Обчислюємо діапазон подання чисел $R = 27 * 11 * 13 * 23 = 88803$.

Нехай $X_1 = 225$, $X_2 = 125$, відповідно, залишки за вибраними модулями рівні $b_1 = 225(\text{mod}27) = 9$, $b_2 = 225(\text{mod}11) = 5$, $b_3 = 125(\text{mod}13) = 8$, $b_4 = 125(\text{mod}23) = 10$.

У вузлі S відбувається об'єднання залишків за формулою (2.1) (рисунок 2.6). Для цього знаходимо базисні числа: $V_1 = 52624$, $V_2 = 80730$, $V_3 = 75141$, $V_4 = 57915$ а $X_S = 15075$.

Відповідно,

$$X1 = 15075 \pmod{23 \cdot 13} = 225,$$

$$X2 = 15075 \pmod{11 \cdot 27} = 125.$$

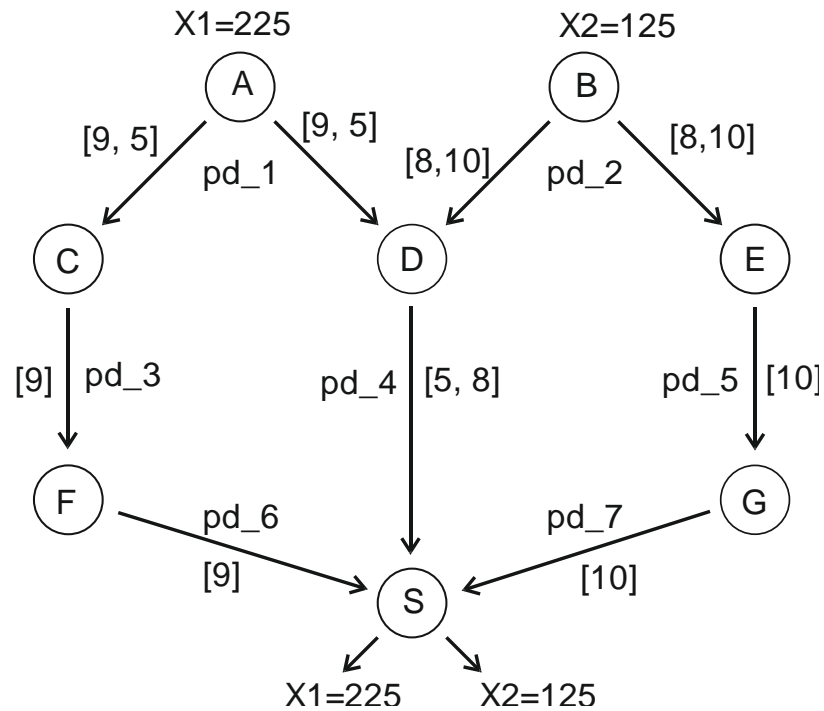


Рисунок 2.6 – Приклад мережного кодування з розподілом залишків для схеми 3

Враховуючи значні затрати енергії на передачу даних в БСМ, важливою є оцінка обсягу даних, які необхідно передавати при використанні різних схем мережного кодування та ретрансляції пакетів. Оцінку обсягу даних, які необхідно передавати при фіксованій розрядності повідомлення, проведемо для розглянутих вище схем мережного кодування 1-3. Нехай джерело інформації генерує повідомлення розрядністю 32 біт.

Визначимо обсяг даних, які необхідно передати з урахуванням ретрансляції пакетів:

– для схеми 1 (див. рисунок 2.2):

$$V1 = k \cdot h \cdot \lceil \log_2 p_i \cdot p_{i+1} \rceil,$$

де k – кількість повідомлень;

h – кількість пакетів, необхідних для передачі повідомлень з вузлів А, В до вузла S;

– для схеми 2 (див. рисунок 2.4)

$$V2 = k \cdot h \cdot \lceil \log_2 p_i \rceil;$$

– для схеми 3 (див. рисунок 2.5)

$$V3 = k \cdot ((h-3) \cdot \lceil \log_2 p_i \rceil + (h-4) \cdot \lceil \log_2 p_i \cdot p_{i+1} \rceil).$$

На рисунку 2.7 показано, як залежить обсяг даних (інформаційних та службових) від кількості повідомлень при різних схемах кодування.

Як видно з рисунку 2.7, запропонована схема 3 мережного кодування (див. рисунок 2.6) для передачі повідомлень вузлами А, В на станцію S потребує передачі, з врахуванням ретрансляції пакетів, на 50% меншого обсягу даних, порівняно із схемою 1 (див. рисунок 2.2), і на 25% меншого обсягу даних, порівняно із схемою 2 (див. рисунок 2.4).

Оцінка загальної пропускної здатності мережі. Для оцінки загальної пропускної здатності мережі розглянемо приклад безпроводної сенсорної мережі, яка налічує $t = 50$ вузлів. Швидкість передачі даних згідно стандарту IEEE 802.15.4 дорівнює 250 кбіт/с. Нехай джерело інформації формує повідомлення, які представляють цілі числа X , де $\lceil \log_2 X_{\max} \rceil = 32$ біт, $\lceil \bullet \rceil$ – округлення до більшого цілого. Для кожного вузла мережі виберемо два взаємно простих модулі з умови $X_{\max} < p_i \cdot p_j$. Модулі для одного вузла мережі виберемо різної розрядності: $\lceil \log_2 p_i \rceil = 22$ біт і $\lceil \log_2 p_{i+1} \rceil = 11$ біт. Необхідна кількість модулів становить $n = 2 \cdot t$. Наприклад, набір взаємно простих модулів для двох вузлів мережі має вигляд [2106421, 2039], [2029, 2116799] – для перших 10 вузлів мережі (таблиця 2.1).

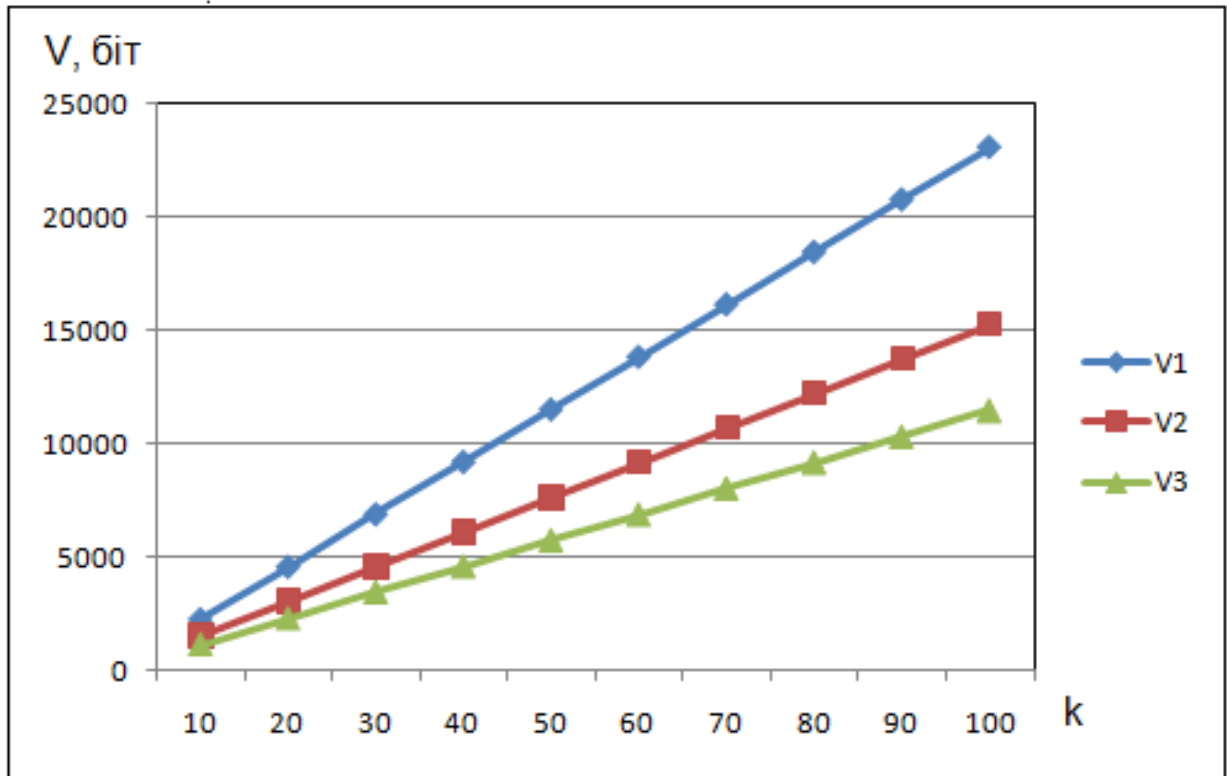


Рисунок 2.7 – Залежність обсягу даних від кількості повідомлень при різних схемах кодування: V1 – схема рисунку 2.2; V2 – схема рисунку 2.4; V3 – схема рисунку 2.5.

Загальну пропускну здатність мережі розрахуємо для схем мережного кодування (1 – 3) (див. рисунки 2.1, 2.3 і 2.5).

Час передачі повідомлень для схеми кодування 1 (див. рисунок 2.1)

$$T_1 = t_{11} + t_{12} + t_{13} + t_{14} + t_{15}, \quad (2.2)$$

де $t_{11} = \frac{\lceil \log_2 b1 \rceil + \lceil \log_2 b2 \rceil}{C}$ – час передачі повідомлень вузлом А (А

→ С, А → D);

t_{12} – час передачі повідомлень вузлом В (В → D, В → E), $t_{12} = t_{11}$;

t_{13} – час передачі повідомлень вузлом С (С → F), вузлом D (D→S), вузлом E (E→G);

t_{14} – час передачі повідомлень вузлом F (F → S);

t_{15} – час передачі повідомлень вузлом G ($G \rightarrow S$).

Таблиця 2.1 – Набір взаємно простих модулів для 10 вузлів мережі

№ вузла	$p1$	$p2$	$\log_2 p1 * p2$
1	2106421	2039	32,000008
2	2116799	2029	32,000006
3	2118889	2027	32,000010
4	2129389	2017	32,000003
5	2135743	2011	32,000004
6	2144269	2003	32,000001
7	2148583	1999	32,000020
8	2150713	1997	32,000002
9	2155057	1993	32,000021
10	2161553	1987	32,000013

Враховуючи, що довжина повідомлень однакова, час передачі повідомлень

$$T_1 = 5 \cdot t_{11},$$

причому час розраховано з врахуванням паралельної передачі пакетів між вузлами $C \rightarrow F$, $D \rightarrow S$, $E \rightarrow G$.

Час передачі повідомлень для схеми кодування 2 (див. рисунок 2.3)

$$T_2 = t_{21} + t_{22} + t_{23} + t_{24} + t_{25}, \quad (2.3)$$

де $t_{21} = \frac{\lceil \log_2 b1 \rceil}{C}$ – час передачі повідомлень вузлом A ($A \rightarrow C$) і вузлом

B ($B \rightarrow D$);

$t_{22} = \frac{\lceil \log_2 b2 \rceil}{C}$ – час передачі повідомлень вузлом A ($A \rightarrow D$), і вузлом B

($B \rightarrow E$);

$$t_{23} = \frac{\lceil \log_2 b2 \rceil + \lceil \log_2 b3 \rceil}{C} - \text{час передачі повідомлень вузлом C (C} \rightarrow \text{F),}$$

D (D \rightarrow S), E(E \rightarrow G);

$$t_{24} - \text{час передачі повідомлень вузлом F (F} \rightarrow \text{S), } t_{24} = t_{21};$$

$$t_{25} - \text{час передачі повідомлень вузлом G (G} \rightarrow \text{S), } t_{24} = t_{22}.$$

Враховуючи, що пакети між вузлами C \rightarrow F, D \rightarrow S, E \rightarrow G передаються паралельно, приймаємо за t_{23} час передачі пакетів (C \rightarrow F, D \rightarrow S, E \rightarrow G). Для даної схеми кодування вибираємо час передавання більшого за розмірами пакету за маршрутом (D \rightarrow S).

Час передачі повідомлень для схеми 3 (див. рисунок 2.5)

$$T_3 = t_{31} + t_{32} + t_{33} + t_{34} + t_{35}, \quad (2.4)$$

$$\text{де } t_{31} = \frac{\lceil \log_2 b1 \rceil + \lceil \log_2 b2 \rceil}{C} - \text{час передачі повідомлень вузлом A (A} \rightarrow$$

C, A \rightarrow D);

$$t_{32} - \text{час передачі повідомлень вузлом B (B} \rightarrow \text{D, B} \rightarrow \text{E);}$$

$$t_{33} = \frac{\lceil \log_2 b2 \rceil + \lceil \log_2 b3 \rceil}{C} - \text{час передачі повідомлень вузлом C (C} \rightarrow \text{F),}$$

вузлом D (D \rightarrow S), вузлом E (E \rightarrow G);

$$t_{34} = \frac{\lceil \log_2 b1 \rceil}{C} - \text{час передачі повідомлень вузлом F (F} \rightarrow \text{S);}$$

$$t_{35} = \frac{\lceil \log_2 b4 \rceil}{C} - \text{час передачі повідомлень вузлом G (G} \rightarrow \text{S).}$$

Оскільки пакети між вузлами C \rightarrow F, D \rightarrow S, E \rightarrow G передаються паралельно, приймаємо за t_{33} час передачі пакетів за маршрутом (D \rightarrow S).

Загальна пропускна здатність мережі з урахуванням тільки часу передачі повідомлень між вузлами для розглянутих схем кодування

$$C = \frac{V}{T} \text{ (біт/с),}$$

де V – обсяг повідомлення;

T – загальний час передачі повідомлень при мережному кодуванні і паралельній передачі повідомлень між вузлами.

На основі проведеного розрахунку часу передачі повідомлень та формули (2.4), побудовано залежності загальної пропускної здатності мережі від розміру повідомлення 8 – 64 біт (рисунок 2.8).

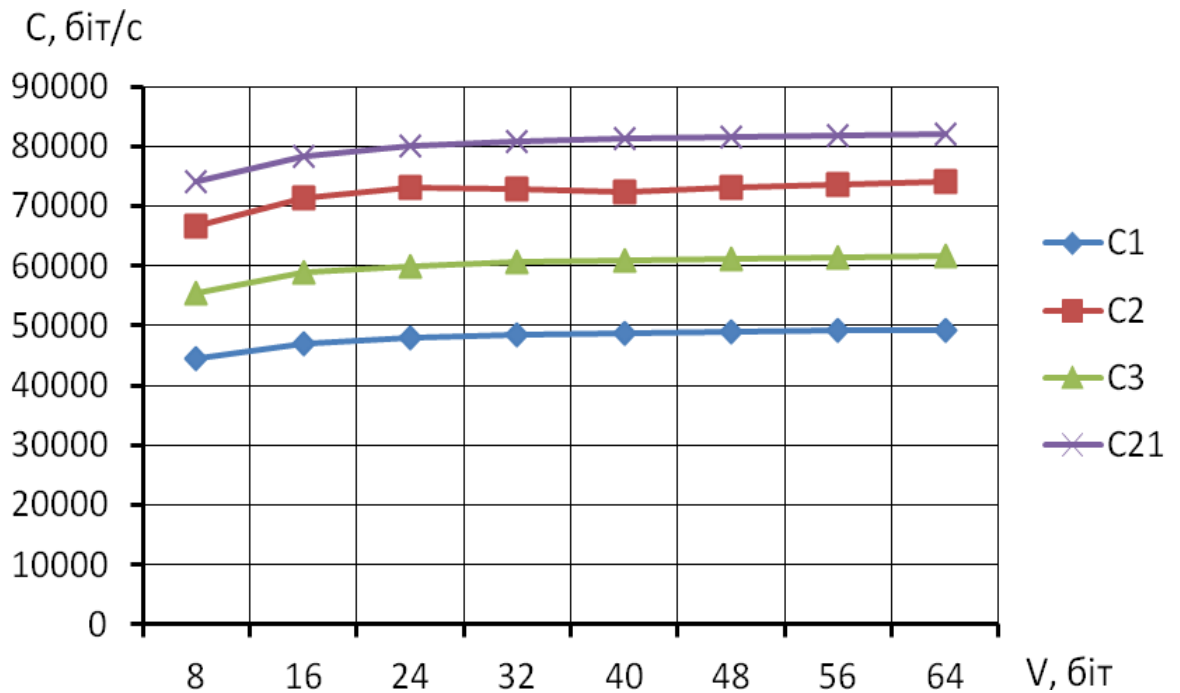


Рисунок 2.8 – Залежність загальної пропускної здатності мережі від розміру повідомлень для різних схем мережного кодування: C1 – для схеми кодування, приведеної на рисунку 2.1; C2 - для схеми кодування, приведеної на рисунку 2.3; C3 – для схеми кодування, приведеної на рисунку 2.5; C21 – для схеми кодування, приведеної на рисунку 2.3 при виборі модулів різної розрядності

Як видно з рисунку 2.8, найбільшу пропускну здатність мережі (C21) забезпечує схема, в якій залишки передаються окремими маршрутами. При цьому вибрані взаємно прості модулі різної розрядності, тому розрядність залишків, які передаються спільним маршрутом, приблизно дорівнює розрядності залишків по окремих маршрутах [95].

Таким чином, в даному підрозділі розроблено та експериментально досліджено метод мережного кодування на основі системи залишкових класів, який забезпечує зменшення обсягу даних, необхідних для відновлення повідомлень на 50%, з урахуванням ретрансляції пакетів. Запропоновано спосіб вибору взаємно простих модулів різної розрядності, тому розрядність залишків, які передаються спільним маршрутом, приблизно дорівнює розрядності залишків по окремих маршрутах. Це дозволило досягти рівномірного розподілу навантаження на канали передачі даних БСМ. Розроблений метод мережного кодування підвищує загальну пропускну здатність мережі за рахунок розділення повідомлення на частини та їх передачі різними маршрутами. На основі отриманих результатів в розділі 4 розроблено протокол мережного кодування для використання в безпроводних сенсорних мережах.

2.2 Кодове розділення каналів зв'язку на основі перетворення системи залишкових класів

Безпроводні сенсорні мережі характеризуються обмеженим ресурсом каналів передачі даних, що пов'язано з вузькою (5 МГц) смугою частотного каналу, використанням неліцензованого діапазону частот (2.4 ГГц) і, як результат, низькою (250 кбіт/с) швидкістю передачі даних. Отже, важливим завданням є ефективне використання наявних ресурсів каналів зв'язку в БСМ. Одним із способів збільшення загальної пропускну здатності є підвищення ефективності розподілу ресурсу між джерелами даних. Для спільного використання ресурсів каналу зв'язку в цифрових системах передачі даних використовують методи розділення (колективного доступу), серед яких найбільш поширеними є частотне, часове та кодове розділення [60].

Оскільки джерелами даних в БСМ є сенсори фізичних параметрів, які формують повідомлення розрядністю 2-3 байти, виникає потреба їх об'єднання з метою підвищення корисної пропускної здатності. Проведені дослідження залежності корисної пропускної здатності від розміру поля даних в пакеті протоколу стандарту IEEE 802.15.4 показали, що при мінімальному розмірі поля даних в структурі протоколу корисна пропускна здатність зменшується в 5 разів. Отже, оптимальною є передача пакетів з полем даних максимальної довжини, в яких відсоток службової інформації по відношенню до корисних даних складає 4% при 80% в пакетах мінімальної довжини [220].

Враховуючи, що до безпроводного модуля може бути підключено в середньому від 4 до 8 сенсорів в [217, 220] запропоновано використання системи залишкових класів для об'єднання даних сенсорів в безпроводному модулі та їх розділення на сервері.

Проведемо оцінку швидкості передачі даних та середнього часу затримки передачі повідомлень при використанні частотного розділення, часового розділення та розділення на основі перетворення системи залишкових класів [218].

Нехай інформація, що вимірюється кожним сенсором збирається в пакети довжиною k біт. При частотному розділенні (FDMA) k - бітні пакети передаються за T секунд по кожному із M каналів, які не перетинаються. Таким чином, швидкість передачі даних дорівнює [63]:

$$R_{FD} = M \cdot \frac{k}{T} . \quad (2.10)$$

При використанні часового розділення (TDMA) кожним джерелом за T/M секунд передається k біт. Відповідно, необхідна швидкість передачі даних дорівнює:

$$R_{TD} = \frac{k}{T/M} . \quad (2.11)$$

При використанні перетворення системи залишкових класів (TRNS) необхідна швидкість передачі даних дорівнює:

$$R_{TRNS} = \frac{M \cdot k}{T} . \quad (2.12)$$

Оскільки рівняння (2.10 – 2.12) ідентичні, то можна записати:

$$R_{FD} = R_{TD} = R_{TRNS} = R = \frac{M \cdot k}{T} . \quad (2.13)$$

Отже, вказані методи розділення вимагають однакової швидкості передачі даних R (біт/с).

Для оцінки середнього часу затримки передачі повідомлень при використанні методів розділення FDMA, TDMA і TRNS припустимо, що при FDMA частотний діапазон розбитий на M ортогональних смуг, при використанні TDMA кадр розділений на M ортогональних часових інтервалів, а при TRNS використовується M взаємно простих модулів. Припустимо, що ресурс каналу використовується на 100%. Відповідно, всі частотні діапазони при FDMA, всі часові інтервали при TDMA і залишки при TRNS будуть заповнені пакетами даних. Не будемо враховувати додаткові затримки, пов'язані з захисними смугами, інтервалами та часом виконання перетворень. В такому випадку час затримки повідомлень дорівнює [60]:

$$D = w + \tau , \quad (2.14)$$

де w – середній час очікування пакета (до передачі),

τ – час передачі пакету.

При FDMA кожний пакет пересилається протягом T секунд, відповідно час передачі пакету для технології FDMA дорівнює:

$$\tau_{FD} = T. \quad (2.15)$$

При використанні TDMA кожний пакет пересилається протягом часового інтервалу T/M секунд. За допомогою рівняння (2.13) час передачі пакету можна виразити

$$\tau_{TD} = \frac{T}{M} = \frac{k}{R}. \quad (2.16)$$

При використанні TRNS всі пакети передаються одночасно, але час передачі окремого пакету також рівний T/M .

$$\tau_{TRNS} = \frac{k}{R}. \quad (2.17)$$

Оскільки канали FDMA доступні постійно, а пакети пересилаються безпосередньо після створення, то час очікування рівний:

$$w_{FD} = 0. \quad (2.18)$$

При використанні TDMA часові інтервали користувачів починаються в різних точках кадру довжиною T секунд. Пакет починає відправлятися через $(m-1) \cdot T/M$ секунд ($1 \leq m \leq M$) після створення пакету. Таким чином, для TDMA середній час очікування пакету перед відправкою дорівнює [60]:

$$w_{TD} = \frac{T}{2} \cdot \left(1 - \frac{1}{M}\right). \quad (2.19)$$

Отже, максимальний час очікування пакету перед відправкою складає $(M - 1) \cdot T / M$ секунд.

При використанні TRNS середній час очікування пакету без врахування часу затраченого на перетворення перед відправкою дорівнює

$$w_{\text{TRNS}} = 0.$$

Отже, середній час затримки передачі повідомлень D_{FD} , D_{TD} і D_{TRNS} при використанні методів розділення FDMA, TDMA і TRNS становить:

$$D_{FD} = T. \quad (2.20)$$

$$D_{TD} = \frac{T}{2} \cdot \left(1 - \frac{1}{M}\right) + \frac{T}{M}. \quad (2.21)$$

$$D_{\text{TRNS}} = \frac{T}{M}. \quad (2.22)$$

Дослідження часу затримки передачі повідомлень для різних методів розділення каналів зв'язку проведено при наступних параметрах: $R = 250$ Кбіт/с, $M = \overline{1,16}$, $k = 500$ біт.

Проведена оцінка середнього часу затримки передачі повідомлень при різних методах розділення каналу зв'язку показала, що використання методу TRNS, який базується на основі перетворення системи залишкових класів, забезпечує менший час затримки передачі повідомлень при однаковій швидкості, порівняно з часовим та частотним розділенням і залежить від кількості сенсорів, дані яких об'єднані в одному пакеті (рисунок 2.9).

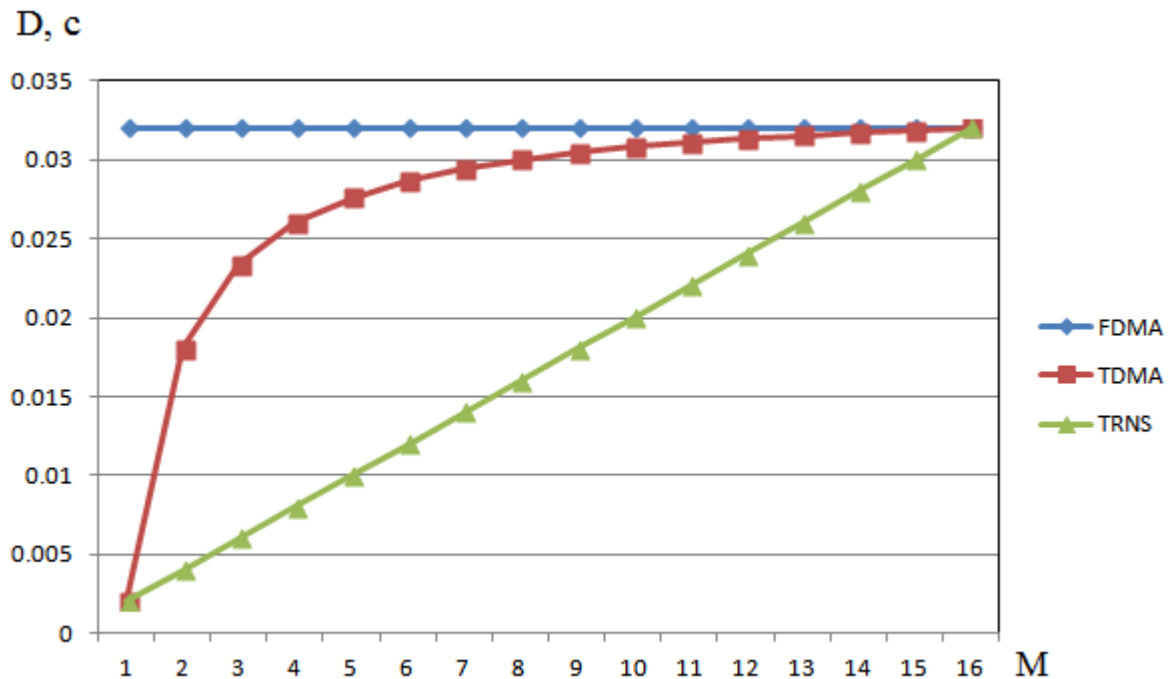


Рисунок 2.9 – Залежність середнього часу затримки повідомлення від кількості каналів: M – кількість частотних каналів при FDMA, кількість часових інтервалів при TDMA і кількість модулів СЗК каналів при TRNS

2.3 Модифікований метод стрибкоподібної зміни частоти

В сучасних розподілених комп'ютерних системах на основі безпроводних каналів зв'язку широко використовуються методи передачі, які базуються на розширенні спектру сигналу [214, 215]. Основна ідея вказаних методів полягає в тому, щоб розподілити інформаційний сигнал по смузі частот набагато більшій за мінімально необхідну. Це значно ускладнює перехоплення сигналів та створення навмисних завад [63].

Системи передачі на основі розширеного спектру сигналу мають наступні переваги [60, 100, 105]:

- забезпечують більш високу завадостійкість в умовах інтенсивних завад;

- дозволяють організувати одночасну роботу багатьох абонентів в загальній смузі частот. Дана властивість використовується в технології мобільного зв'язку, відомій як багаточисельний доступ з кодовим розділенням (code division multiple access – CDMA);

- дозволяють реалізувати асинхронно-адресні принципи системи зв'язку і кодове розділення абонентів;

- забезпечують ефективніше використання спектру частот на обмеженій території порівняно з вузькосмуговими системами зв'язку;

- дозволяють приховувати і шифрувати сигнали. Відновити зашифровані дані зможе тільки користувач, якому відомий код розширення.

В стандарті IEEE 802.11 визначено два методи передачі даних на основі розширення спектру. Це метод прямої послідовності (direct sequence spread spectrum – DSSS) та метод стрибкоподібної зміни частоти (frequency-hopping spread spectrum FHSS) [60, 63].

При розширенні спектру методом прямої послідовності кожному вхідному біту ставиться у відповідність псевдовипадкова послідовність. В якості псевдовипадкової послідовності використовують послідовності максимальної довжини (M - послідовності), які генеруються регістром зсуву із зворотнім зв'язком.

При використанні розширення спектру зі стрибкоподібною зміною частоти передача сигналу відбувається з допомогою визначеного набору несучих частот, які змінюються через певні інтервали часу за псевдовипадковим законом. Для передачі даних використовується 2^k несучих частот. Відстань між несучими частотами вибирається рівною ширині смуги вхідного сигналу. Кожний канал використовується фіксований інтервал часу, протягом якого відбувається передача визначеної кількості бітів. В залежності від часу переключення несучих частот розрізняють метод розширення спектру з повільною стрибкоподібною зміною частоти або метод зі швидкою зміною частоти.

Менш поширеним є метод переключення часових інтервалів (time hopping – TH), який використовується при наявності навмисних перешкод, оскільки він приховує координати сигналу.

Безпроводні сенсорні мережі працюють в неліцензованих частотних діапазонах, тому високою є ймовірність спотворення даних, викликана перешкодами, що створюються сторонніми джерелами радіосигналів. Використання методу стрибкоподібної зміни частоти та систематичних коректуючих кодів не дозволяє в повній мірі вирішити проблему підвищення надійності передачі даних на фізичному рівні безпроводних сенсорних мереж, оскільки можливе пошкодження даних, які передаються на окремій несучій частоті.

Розроблений метод передачі даних базується на розширенні спектру сигналу методом стрибкоподібної зміни частоти та перетворенні системи залишкових класів (СЗК) [80, 89, 96]. Двійковий код числа N , яке підлягає передачі, подається у вигляді залишків b_i у вибраній системі модулів $(p_1, p_2, \dots, p_{k-1}, p_k)$:

$$N = \{b_1, b_2, \dots, b_i, \dots, b_k\}.$$

Отримані залишки передаються за допомогою визначених наборів частот $f_1 \dots f_n$, вибір яких здійснюється за псевдовипадковим законом (рисунок 2.10).

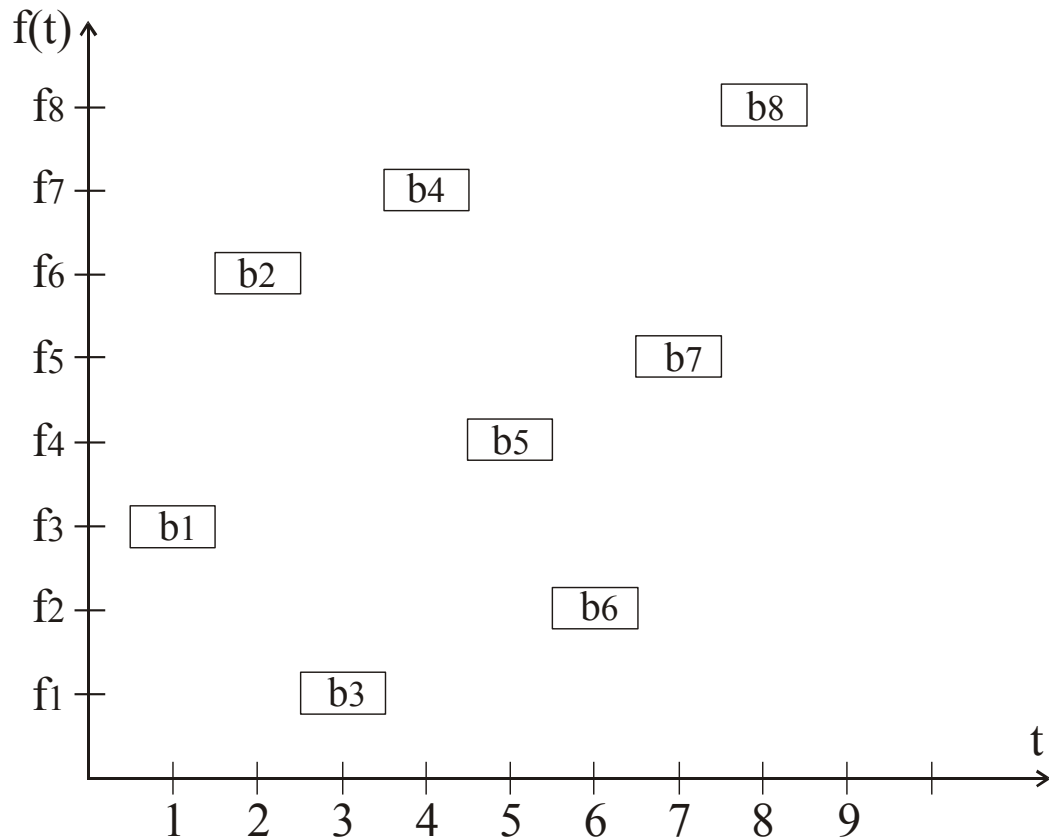


Рисунок 2.10 – Передача залишків методом стрибкоподібної зміни частоти

Для генерування псевдовипадкових чисел вибрано метод лінійного порівняння. Послідовність псевдовипадкових чисел $\{X_n\}$ отримаємо за допомогою рівняння:

$$X_{n+1} = (a \cdot X_n + c) \pmod{m},$$

де m – модуль, $m > 0$;

a – множник, $0 \leq a < m$;

c – інкремент, $0 \leq c < m$;

X_0 – початкове значення, $0 \leq X_0 < m$.

Якщо параметри m , a , c і X_0 – цілі числа, то результатом буде послідовність цілих чисел, кожне з яких належить діапазону $0 \leq X_n < m$.

Обчислені залишки передаються на вибраних несучих частотах (див.рисунок 2.10) [191].

Зміна частоти сигналу відбувається через визначені інтервали часу. Зміна робочих частот приймача синхронізується з передавачем, що дозволяє прийняти сигнал.

Відстань між несучими частотами дорівнює ширині смуги вхідного сигналу. При передачі кожний канал використовується фіксований інтервал часу. Протягом цього часу відбувається передача визначеної кількості бітів, що відповідає розрядності залишку b_i . Черговість використання каналів задається псевдовипадковою послідовністю, а оскільки приймач і передавач використовують одну і ту саму послідовність чисел, то зміна каналів відбувається синхронно.

Наприклад, при діапазоні вхідних чисел від $0 \leq N < 2^{32}$ вибираємо наступні модулі $p_1 = 5$, $p_2 = 7$, $p_3 = 13$, $p_4 = 17$, $p_5 = 19$, $p_6 = 23$, $p_7 = 29$, $p_8 = 31$. Знайдемо залишки b_i для числа $N = 135$ за відповідними модулями $b_1 = 0$, $b_2 = 2$, $b_3 = 5$, $b_4 = 16$, $b_5 = 2$, $b_6 = 20$, $b_7 = 19$, $b_8 = 11$. Виберемо $k = 8$ несучих частот, які складають 8 каналів (рисунок 2.10). Отримані залишки передаються у вибраній системі частот, зміна яких відбувається за псевдовипадковим законом.

В приймачі відбувається зворотне перетворення даних, поданих у системі залишкових класів, в позиційну систему числення.

Щоб забезпечити можливість виявлення та виправлення помилок в процесі передачі даних, необхідно розширити діапазон модулів системи залишкових класів $(p_1, p_2, \dots, p_{k-1}, p_k)$ до $(p_1, p_2, \dots, p_{k-1}, p_k, p_{k+1}, p_{k+2})$ або зменшити діапазон подання чисел до $(p_1, p_2, \dots, p_{k-2})$.

Структурна схема системи передачі на основі запропонованого методу перетворення та передачі даних подана на рисунку 2.11.

Дані в коді NRZ поступають на кодер, на виході якого одержуємо залишки за відповідними модулями. Залишки b_i поступають на вхід модулятора 2, з синтезатора частот 3 на модулятор поступає несуча частота. Значення несучої частоти задає генератор псевдовипадкових чисел 4.

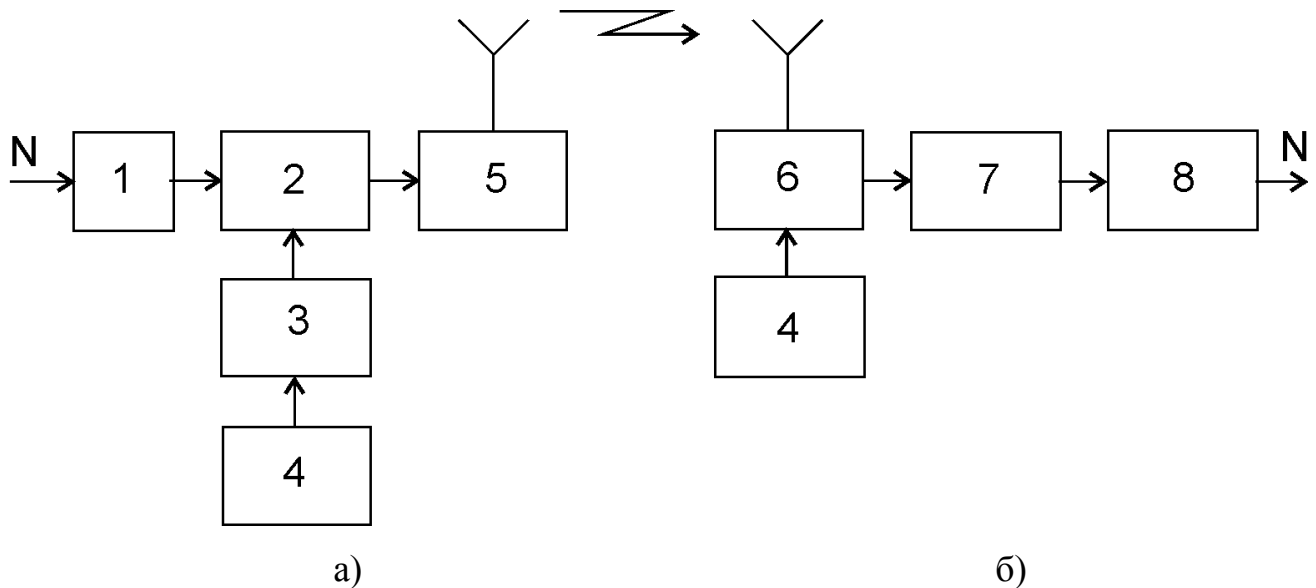


Рисунок 2.11 – Структурна схема системи передачі даних: а) передавач, б) приймач: 1 – кодер; 2 – модулятор; 3 – синтезатор частот; 4 – генератор псевдовипадкових чисел; 5 – передавач; 6 – приймач; 7 – демодулятор; 8 – декодер.

На приймальній стороні сигнал поступає на демодулятор 7, вибір частоти якого здійснюється за псевдовипадковим законом. В декодері 8 відбувається зворотне перетворення системи залишкових класів в позиційну систему, тобто відновлення значення N , перевірка на наявність помилок і їх виправлення.

Враховуючи, що в канал зв'язку передається не двійковий код, а залишки від ділення даного коду на відповідні модулі і вибір несучої частоти здійснюється за псевдовипадковим законом, то дані є додатково захищені від несанкціонованого доступу, а створення спланованих завад на одній із несучих частот призведе до спотворення тільки залишку за відповідним

модулем, що виявляється та виправляється введенням двох додаткових модулів [89].

Розроблений модифікований метод передачі даних забезпечує підвищення надійності передачі даних на фізичному рівні БСМ та захист від спланованих завад за рахунок розділення повідомлення на залишки та їх передачі методом стрибкоподібної зміни частоти. Так як вибір несучої частоти здійснюється за псевдовипадковим законом, то створення спланованих завад на одній із несучих частот призведе до спотворення залишків тільки за одним модулем, що можна виявити за допомогою розширеної системи модулів.

2.4 Перетворення та обробка зображень в безпроводних сенсорних мережах

Використання методів стиснення даних в БСМ дозволяє ефективно використовувати обмежену пропускну здатність каналів зв'язку та зменшити енергозатрати на передачу даних.

На даний час розроблена велика кількість методів та алгоритмів стиснення зображень, серед яких найбільш поширеним є JPEG, який вже реалізовано в більшості мікросхем CMOS камер [9, 13]. Недоліком існуючих алгоритмів стиснення зображення є їх послідовна робота, що особливо помітно при їх реалізації на пристроях з обмеженими апаратними ресурсами [175].

Алгоритми стиснення зображень для використання в безпроводних сенсорних мережах візуального моніторингу об'єктів повинні мати наступні характеристики: забезпечувати заданий коефіцієнт стиснення, низьке

енергоспоживання, малу обчислювальну складність, низькі вимоги до пам'яті, можливість вбудованої реалізації.

В загальному випадку під кодуванням розуміється процес перетворення вхідних даних в форму, зручну для безпосередньої передачі, обробки та зберігання. В даному пункті під кодування розуміється перетворення пікселів зображення у систему залишкових класів.

При переході до представлення чисел в СЗК отримуємо незалежні числа малої розрядності (2 – 8 біт), що дозволяє підвищити швидкодію виконання арифметичних операцій.

Кольорове зображення в форматі RGB подається у вигляді масиву кольорових пікселів $M \times N \times 3$, де M – кількість рядків, N – кількість стовбців, причому кожен такий піксель є триплетом, елементи якого відповідають трьом колірним компонентам: червоному, зеленому і синьому.

Найбільш поширеним варіантом є подання кожної компоненти восьми бітовим значенням, відповідно, RGB зображення має глибину 24 біта [9, 12, 13].

Запропонований метод базується на переході від подання пікселів зображення в двійковій системі числення до їх подання в системі залишкових класів [81, 94]. Оскільки пікселі зображення приймають значення в діапазоні від 0 до 255, то в СЗК необхідно вибрати взаємно прості модулі добутку, який буде більший 255.

Отже, вибираємо наступні набори модулів: $\{p_1 = 3, p_2 = 7, p_3 = 13\}$, де $P = 3 \times 7 \times 13 = 273$ або $\{p_1 = 5, p_2 = 7, p_3 = 8\}$, $P = 5 \times 7 \times 8 = 280$. Для подальшої роботи оберемо модулі $\{5, 7, 8\}$, тому що залишки за приведеними модулями в двійковому коді мають однакову розрядність $m = 3$.

З цифрової камери компоненти зображення RGB поступають на перетворювач з двійкового коду в код СЗК, на виході якого отримуємо залишки b_i від ділення значень пікселів на вибрану систему модулів p_i (рисунок 2.12) [81]. Отримані масиви залишків поступають на кодер, який

здійснює подальшу обробку (стиснення) зображення, поданого залишками з використанням алгоритму арифметичного кодування (рисунок 2.13) [98]. З виходу кодерів стиснуті послідовності надходять на передавальні пристрої та передаються паралельними каналами [97, 107, 216].

$$A = \begin{vmatrix} 95 & 111 & 119 & 125 & 132 & 130 \\ 95 & 111 & 120 & 127 & 134 & 130 \\ 95 & 112 & 121 & 130 & 134 & 128 \\ 100 & 115 & 122 & 131 & 135 & 130 \\ 105 & 118 & 124 & 132 & 137 & 132 \\ 104 & 119 & 124 & 131 & 137 & 135 \end{vmatrix}$$

$$\begin{array}{ccc} \begin{array}{c} A \pmod{5} \\ \begin{vmatrix} 0 & 1 & 4 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 & 4 & 0 \\ 0 & 2 & 1 & 0 & 4 & 3 \\ 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 3 & 4 & 2 & 2 & 2 \\ 4 & 4 & 4 & 1 & 2 & 0 \end{vmatrix} \end{array} & \begin{array}{c} A \pmod{7} \\ \begin{vmatrix} 4 & 6 & 0 & 6 & 6 & 4 \\ 4 & 6 & 1 & 1 & 1 & 4 \\ 4 & 0 & 2 & 4 & 1 & 2 \\ 2 & 3 & 3 & 5 & 2 & 4 \\ 0 & 6 & 5 & 6 & 4 & 6 \\ 6 & 0 & 5 & 5 & 4 & 2 \end{vmatrix} \end{array} & \begin{array}{c} A \pmod{8} \\ \begin{vmatrix} 7 & 7 & 7 & 5 & 4 & 2 \\ 7 & 7 & 0 & 7 & 6 & 2 \\ 7 & 0 & 1 & 2 & 6 & 0 \\ 4 & 3 & 2 & 3 & 7 & 2 \\ 1 & 6 & 4 & 4 & 1 & 4 \\ 0 & 7 & 4 & 3 & 1 & 7 \end{vmatrix} \end{array} \end{array}$$

Рисунок 2.12 – Подання даних в СЗК

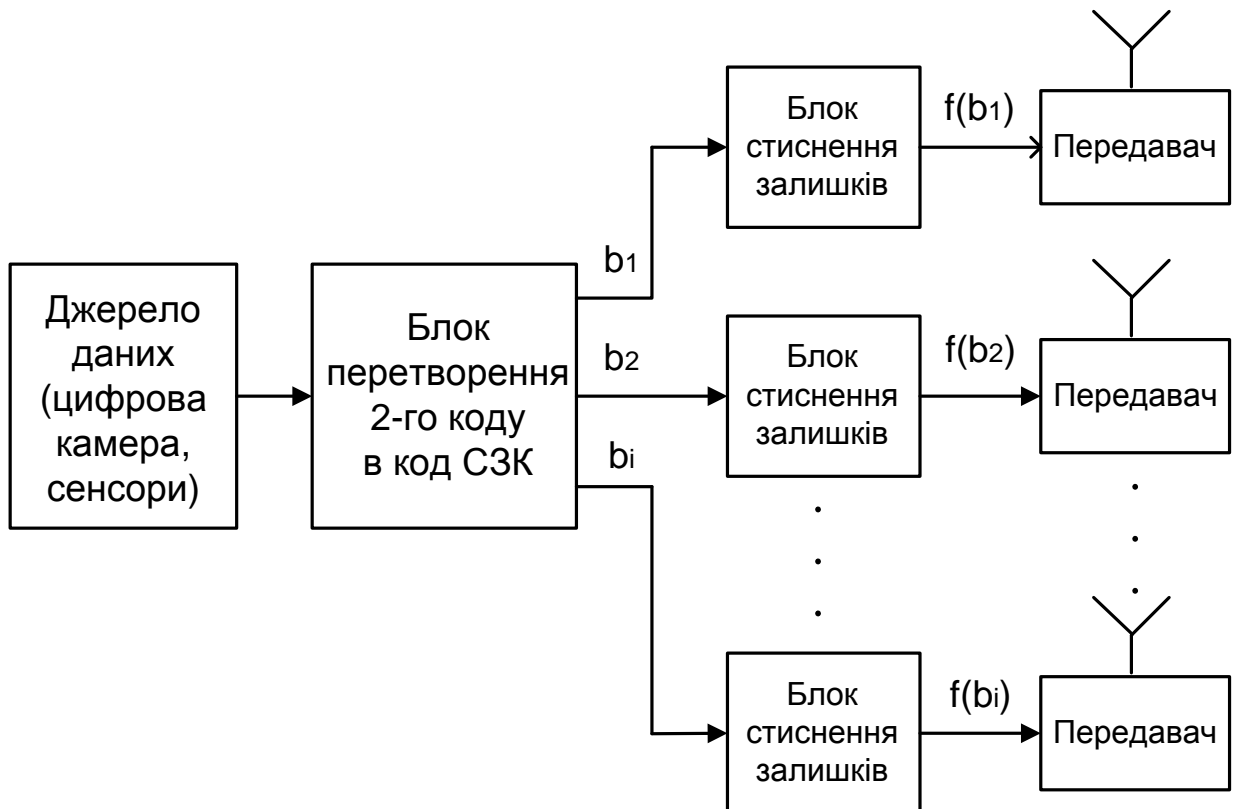


Рисунок 2.13 – Схема перетворення та обробки зображення

В декодері відбувається відновлення повідомлень з використанням алгоритму арифметичного декодування. Після відновлення залишків в декодері вони надходять на блок перетворення з коду СЗК в двійковий код (рисунок 2.14).

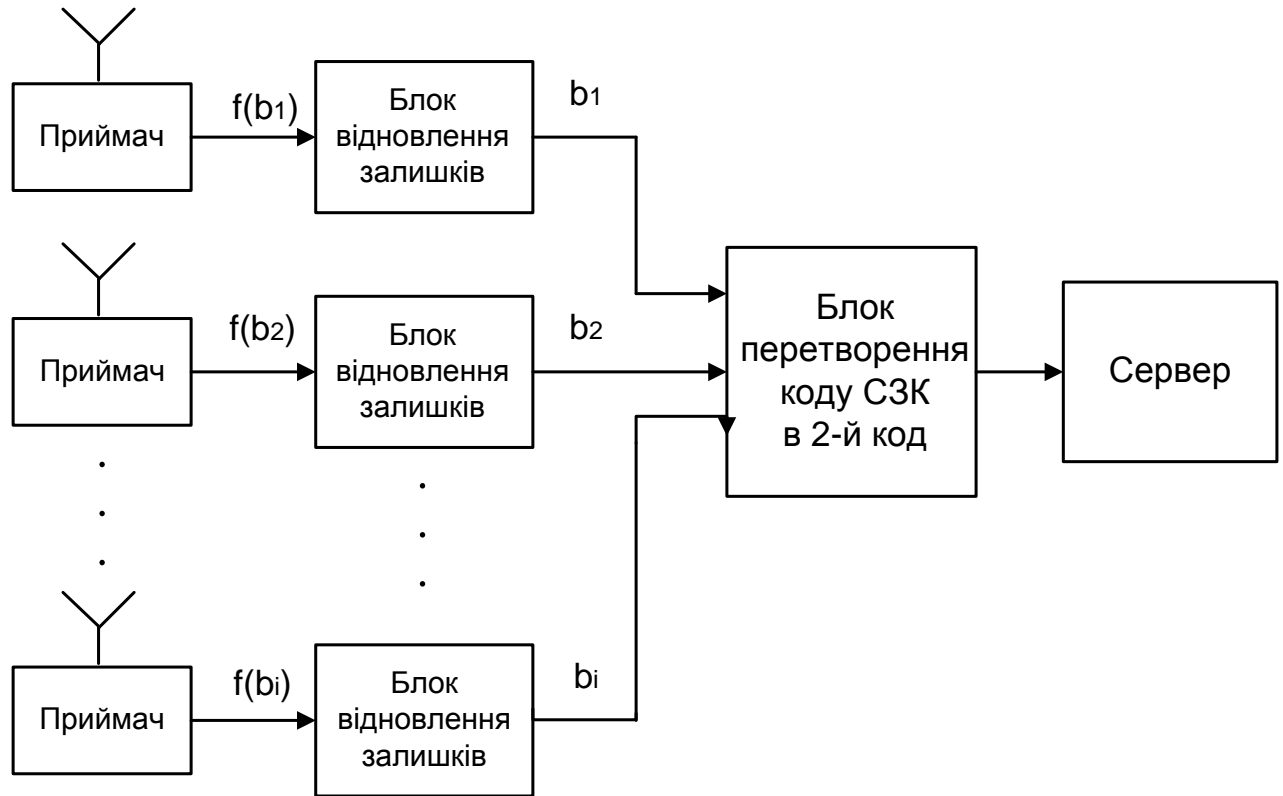


Рисунок 2.14 – Схема приймання та відновлення зображення

Перетворення з СЗК в двійкову систему числення відбувається за формулою [2]:

$$A = \left(\sum_{i=1}^3 b_i \cdot B_i \right) \bmod P,$$

де B_i – ортогональні базиси, $P = \prod_{i=1}^3 p_i$, p_i – взаємно прості числа, 3 – кількість модулів, $i = \overline{1, 3}$,

$$B_i = \frac{P}{p_i} \cdot \delta_i \equiv 1 \pmod{p_i},$$

де $0 < \delta_i < p_i$ – вага ортогонального елемента.

Для модулів 5, 7, 8 ортогональні базиси дорівнюють: $B_1 = 56$, $B_2 = 120$, $B_3 = 105$.

Враховуючи, що зворотне перетворення СЗК відновлює вхідні дані без втрат, відповідно запропонований метод забезпечує зменшення надлишковості зображень без втрат якості (рисунок 2.15).

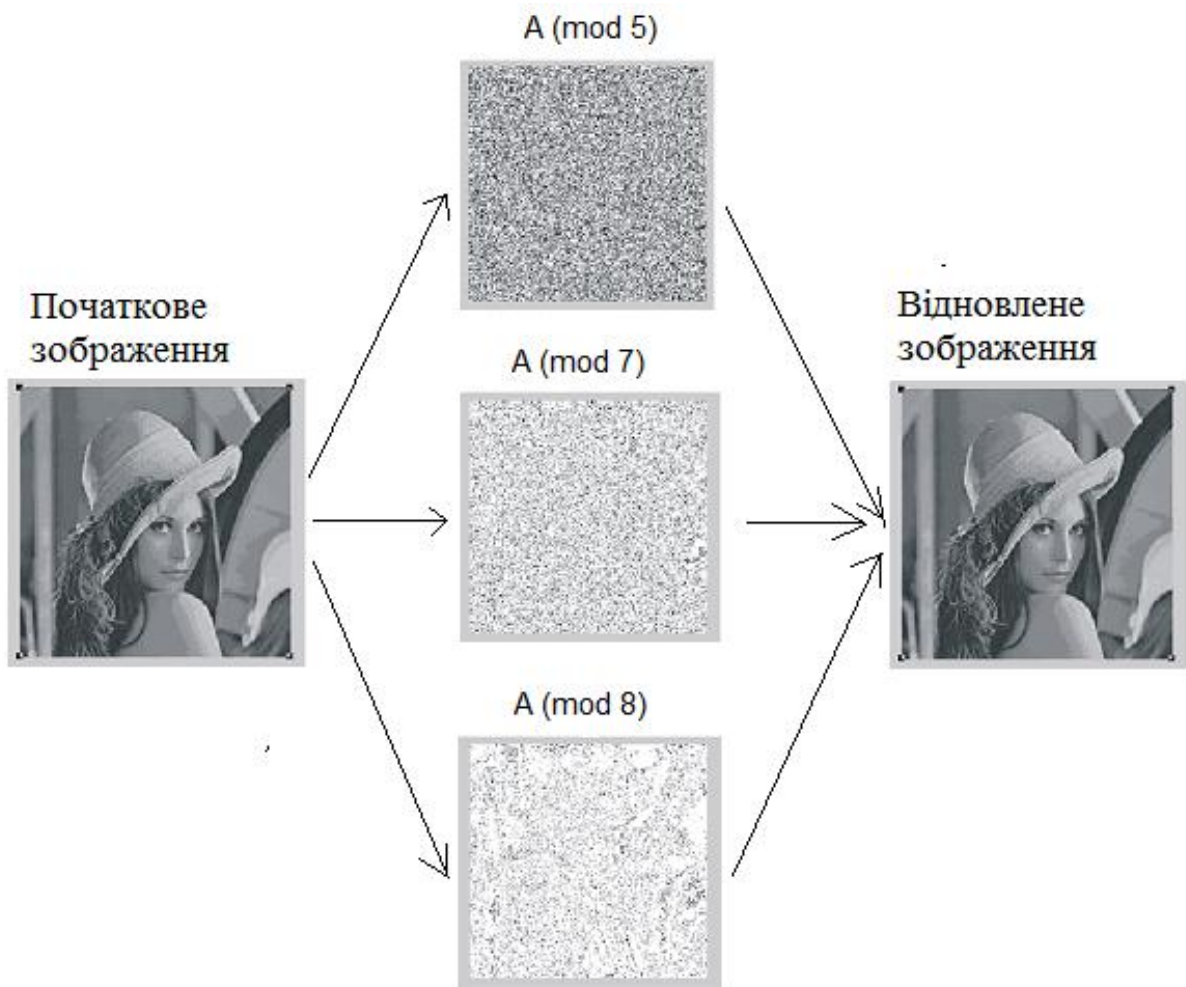


Рисунок 2.15 – Приклад перетворення зображення

Дослідження коефіцієнту стиснення частин зображень, поданих залишками, проводилися на тестових зображеннях та файлах інших типів. Результати досліджень наведені в таблиці 2.2.

Алгоритм дослідження складається з наступних кроків.

1. Дані вхідного файлу зчитуються і перетворюються в тип *byte* (мова *java*).
2. З кожного байту даних обчислюються залишки за модулями 5, 7, 8.
3. Отримані залишки за модулями записуються в окремі файли.
4. Обробка залишків з використанням алгоритмів стиснення даних.
5. Передача стиснутих залишків.
6. Розархівування отриманих даних.
7. Відновлення позиційного подання числа з використанням зворотного перетворення СЗК.
8. Перетворення даних в заданий тип.

Таблиця 2.2 – Стиснення даних в системі залишкових класів

Назва і тип файлу	Розмір файлу, байт	Розмір файлів із залишками за модулями, байт			Розмір стиснутих файлів із залишками, байт			Коефіцієнт стиснення, %
		p_1	p_2	p_3	p_1	p_2	p_3	
laptop.jpg	1664005	1173641	1219937	1237525	431314	487334	514055	13,9
intel.jpg	398977	276539	283346	292528	102439	116094	121131	14,8
imagePi.jpg	206281	137331	140767	143709	51596	58127	60939	17,3
lena.tiff	786572	517321	531654	534935	196056	222433	232466	17,2
numb512.tiff	262278	156115	159553	160167	60434	68516	71658	23,5
monarch.tif	1179784	680073	689737	690486	266415	301166	315092	25,2
info.doc	61440	27543	27649	27327	8795	9657	9434	54,6
sensor1.txt	152332	58589	58589	58589	13978	15654	16911	69,4
sensor2.txt	1812992	697304	697304	697304	167502	191626	207780	68,7
sensor.zip	82500	58813	61517	62226	21413	24447	25967	12,9

За рахунок обробки даних малої розрядності (розрядність залишків становить 3 біти) та паралельної роботи блоків стиснення залишків за вибраними модулями СЗК отримуємо підвищення швидкодії стиснення зображення в 2-3 рази (таблиця 2.3).

Проведені експериментальні дослідження підтвердили переваги запропонованого методу перетворення зображень в СЗК з подальшим стисненням залишків методом арифметичного кодування. Результати досліджень показали, що запропонований метод стиснення забезпечує коефіцієнт стиснення в діапазоні від 13% до 24 % і залежить від формату зображення та від 54,6% до 69% для файлів з розширенням txt та doc.

Експериментальне дослідження часу стиснення вхідного файлу та файлів залишків наведено в таблиці 2.3.

Таблиця 2.3 – Час стиснення вхідного файлу та файлів залишків

Назва і тип файлу	Розмір файлу, байт	Час стиснення вхідного файлу, мс	Час стиснення файлу залишків, мс	Підвищення швидкодії, разів
laptop.jpg	1664005	2217	852	2,60
intel.jpg	398977	557	226	2,46
imagePi	206281	318	146	2,18
lena.tiff	786572	1038	572	1,81
numb512.tiff	262278	342	154	2,22
monarch.tif	1179784	1299	407	3,19
info.doc	61440	90	46	1,96
sens_1.txt	152332	147	59	2,49
sens2.txt	1812992	1003	582	2,24
sensor.zip	82500	146	65	2,25

Таким чином, запропонований метод обробки та передачі зображень забезпечує зменшення часу стиснення файлу в 2-3 рази за рахунок поділу зображення на модулі системи залишкових класів та паралельного стиснення отриманих залишків. Застосування арифметичного кодування для стиснення залишків забезпечує коефіцієнт стиснення без втрат від 13% до 24% і залежить від формату зображення та від 55% до 69% для файлів з розширенням txt та doc.

Використання розробленого методу в безпроводних сенсорних мережах дозволить зменшити затримку при передачі зображень за рахунок передачі стиснутих залишків по різних маршрутах.

Висновки до другого розділу

1. Розроблений метод мережного кодування на основі системи залишкових класів забезпечує зменшення обсягу даних на 50% з врахуванням ретрансляції пакетів, які необхідні для відновлення повідомлень, та підвищує загальну пропускну здатність мережі приблизно на 60%. Запропонований спосіб вибору взаємно простих модулів, при якому модулі вибираються різної розрядності, тому розрядність залишків, які передаються через спільний маршрут, приблизно дорівнює розрядності залишків на окремих маршрутах.

2. Проведено дослідження швидкості передачі даних та середнього часу затримки передачі повідомлень при розділенні ресурсів каналу зв'язку з використанням методів частотного, часового та кодового розділення на основі системи залишкових класів, яке показало, що використання методу кодового розділення на основі системи залишкових класів забезпечує менший час затримки передачі повідомлень при однаковій швидкості,

порівняно з часовим і частотним розділенням, і залежить від кількості даних сенсорів, об'єднаних в спільному пакеті.

3. Розроблено модифікований метод стрибкоподібної зміни частоти, який забезпечує підвищення надійності передачі даних на фізичному рівні безпроводних мереж та захист від спланованих завад за рахунок розділення повідомлення на залишки та їх передачі методом стрибкоподібної зміни частоти. Оскільки вибір несучої частоти здійснюється за псевдовипадковим законом, то створення спланованих завад на одній із несучих частот призведе до спотворення залишків тільки за одним модулем, що можна виявити за допомогою розширеної системи модулів.

4. Запропонований метод перетворення та обробки зображень забезпечує зменшення часу обробки зображення в 2–3 рази за рахунок поділу зображення на модулі системи залишкових класів та паралельного стиснення отриманих залишків. Застосування арифметичного кодування для стиснення залишків забезпечує коефіцієнт стиснення без втрат від 13% до 24 % і залежить від формату зображення, та від 54.6% до 69% для файлів з розширенням txt та doc.

РОЗДІЛ 3

ТЕОРЕТИЧНІ ОСНОВИ ПІДВИЩЕННЯ НАДІЙНОСТІ ПЕРЕДАЧІ ДАНИХ У БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖАХ

3.1 Коректуючі коди на основі модулярної арифметики

Проведений в першому розділі аналіз механізмів контролю помилок показав, що для підвищення надійності роботи БСМ необхідна розробка нового класу коректуючих кодів, які б забезпечували можливість гнучкого вибору параметрів кодів та реалізацію адаптивних схем контролю помилок.

В запропонованих коректуючих кодах на основі модулярної арифметики пакет даних в двійковому коді, який підлягає передачі, розбивається на k блоків однакової довжини (тетради або байти) [86, 102, 224]:

$$a_j^1 \dots a_3^1 a_2^1 a_1^1 a_0^1, a_j^2 \dots a_3^2 a_2^2 a_1^2 a_0^2, a_j^i \dots a_3^i a_2^i a_1^i a_0^i, \dots, a_j^k \dots a_3^k a_2^k a_1^k a_0^k,$$

де a^i – розряд даних в двійковому коді, $j = 4, 8$.

При цьому значення контрольного розряду дорівнює

$$x_{k+1} = |v_1 \cdot x_1 + v_2 \cdot x_2 + \dots + v_i \cdot x_i + \dots + v_k \cdot x_k|_p, \quad (3.1)$$

де v_i – коефіцієнти, взаємно прості з p ;

x_i – байт даних в двійковій або десятковій системі числення, причому

$$x_i = a_7^i \dots a_3^i a_2^i a_1^i a_0^i = a_7 \cdot 2^7 + \dots + a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0.$$

Припустимо, що в процесі передачі відбулась помилка в одному з блоків даних (спотворення зазнали від 1 до 8 двійкових розрядів) і замість числа X отримано X'

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

Значення контрольного розряду по отриманих даних становить

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

Синдром обчислюємо, як різницю між розрахованим і прийнятим контрольними символами:

$$\delta = \left| x'_{k+1} - x_{k+1} \right|_P. \quad (3.2)$$

Рівняння (3.2) запишемо у вигляді

$$\begin{aligned} & \left| v_1 \cdot x'_1 + v_2 \cdot x'_2 + \dots + v_i \cdot x'_i + \dots + v_k \cdot x'_k \right|_P - \\ & \left| v_1 \cdot x_1 + v_2 \cdot x_2 + \dots + v_i \cdot x_i + \dots + v_k \cdot x_k \right|_P = \delta', \end{aligned} \quad (3.3)$$

Оскільки при $\delta = 0$ – помилка відсутня, то рівняння (3.3) можна записати:

$$\begin{aligned} & \left| v_1 \cdot x'_1 + v_2 \cdot x'_2 + \dots + v_i \cdot x'_i + \dots + v_k \cdot x'_k \right|_P - \\ & \left| v_1 \cdot x_1 + v_2 \cdot x_2 + \dots + v_i \cdot x_i + \dots + v_k \cdot x_k \right|_P = 0', \end{aligned}$$

або

$$\left| v_1 \cdot (x'_1 - x_1) + v_2 \cdot (x'_2 - x_2) + \dots + v_i \cdot (x'_i - x_i) + \dots + v_k \cdot (x'_k - x_k) \right|_P = 0. \quad (3.4)$$

Припустивши, що помилка відбулася в одному інформаційному символі, рівняння (3.4) набуде вигляду:

$$\left| v_i \cdot (x'_i - x_i) \right|_P = 0, \quad (3.5)$$

Отже, з рівняння (3.5) видно, що тільки при $x'_i = x_i$ синдром дорівнює нулю ($\delta = 0$).

Для виявлення помилки в будь-якому з блоків необхідно, щоб кожному значенню δ_i відповідало одне значення помилки e_i . Для виконання даної умови коефіцієнти лінійної форми повинні бути взаємно прості з P , а значення контрольного модуля повинно дорівнювати $P \geq 2^m$, де m – кількість двійкових розрядів у вибраному блоці даних (при розбитті пакету даних на байти $m = 8$).

Для оцінки ефективності коректуючого коду визначимо зв'язок між надлишковістю і можливістю виявляти та виправляти помилки.

Мінімальна кодова відстань модулярного коректуючого коду $d_{\min} = n - k + 1$, де n – загальна довжина коду, k – кількість інформаційних блоків. Отже, розроблений коректуючий код виявляє всі сукупності із t або меншої кількості помилок при умові, що $d_{\min} \geq t + 1$.

Розглянемо приклад виявлення помилок з використанням коректуючого коду на основі модулярної арифметики. В якості контрольного модуля виберемо $P = 16$; коефіцієнти лінійної форми: $v_1 = 5$, $v_2 = 7$, $v_3 = 11$, інформаційні блоки даних в десятковій системі числення: $x_1 = 2$, $x_2 = 10$, $x_3 = 9$. Обчислимо значення контрольного розряду $x_{k+1} = |5 \cdot 2 + 7 \cdot 10 + 11 \cdot 9|_{16} = 3$; введемо помилку в першому блоці $x'_1 = 4$. Значення контрольного розряду з врахуванням помилки дорівнює $x'_{k+1} = |5 \cdot 4 + 7 \cdot 10 + 11 \cdot 9|_{16} = 13$. Відповідно, $\delta = |x'_{k+1} - x_{k+1}|_{16} = |13 - 3|_{16} = 10$, отже, помилка виявлена.

Проведені експерименти для значення контрольного модуля $P \in 2^3 \div 2^8$ показали, що розроблений модулярний коректуючий код (М – код) з одним контрольним модулем забезпечує 100 % виявлення помилок в одному блоці даних та 87% в двох блоках даних, при цьому коректуючі можливості залежать також від значення контрольного модуля.

3.1.1 Виправлення однократних помилок

Для виправлення помилок використаємо поняття альтернативної множини [3]. Альтернативна множина для розроблених коректуючих кодів – це множина блоків даних [86]

$$\Theta = (x_{\delta}^1, x_{\delta}^2, \dots, x_{\delta}^k),$$

причому помилка в будь-якому із розрядів може призвести до зміни значення синдрому – δ_i в межах умови $0 < \delta_i < p$. Для кожного блоку x_i можна визначити значення помилки $e_{x_i}^i$ (таблиця 3.1) і, відповідно, значення синдрому [152]

$$\delta = \left| x'_{k+1} - x_{k+1} \right|_p = \left| v_1 \cdot x_1 + v_2 \cdot x_2 + \dots + v_i \cdot x'_i + \dots + v_k \cdot x_k \right|_p - \left| v_1 \cdot x_1 + v_2 \cdot x_2 + \dots + v_i \cdot x_i + \dots + v_k \cdot x_k \right|_p. \quad (3.6)$$

При виникненні помилки в блоці даних значення $x'_i = x_i + e_i$, де e_i – значення помилки, яке залежить від розрядності даних і дорівнює: $-(2^m - 1) \leq e_i \leq 2^m - 1$.

Таблиця 3.1 – Значення помилки для кожного блоку даних

Значення синдрому δ	Блок даних x_i					
	x_1	x_2	...	x_i	...	x_k
1	$e_{x_1}^1$	$e_{x_2}^1$...	$e_{x_i}^1$...	$e_{x_n}^1$
2	$e_{x_1}^2$	$e_{x_2}^2$...	$e_{x_i}^2$...	$e_{x_n}^2$
...
$P-1$	$e_{x_1}^{P-1}$	$e_{x_2}^{P-1}$...	$e_{x_i}^{P-1}$...	$e_{x_n}^{P-1}$

Коефіцієнти лінійної форми знаходимо відповідно до умови: при виникненні будь-якої помилки в одному з блоків даних значення синдрому повинно відрізнятись.

При помилці в блоці даних x_i за формулою (3.6) знаходимо синдром

$$\delta_i = |v_i \cdot e_{x_i}|_P.$$

Для однозначного виправлення однократної помилки (помилки в одному блоці даних) необхідно, щоб для кожного значення синдрому альтернативна множина Θ складалася із одного блоку даних. Значення контрольного модулю вибираємо із умови [87]:

$$p \geq 2 \cdot k \cdot (2^m - 1), \quad (3.7)$$

де m – розрядність блоку даних, k – кількість блоків даних.

Розглянемо на прикладі локалізацію помилок. Нехай інформаційне повідомлення складається з трьох блоків даних, тобто $n = 3$, при цьому розрядність кожного блоку рівна $m = 3$ біти.

Взаємно прості коефіцієнти знаходимо згідно розробленого алгоритму та програмного забезпечення (додаток Б): $v_1 = 54$, $v_2 = 25$, $v_3 = 67$. Мінімальне теоретичне значення контрольного модуля згідно (3.7): $P \geq (2 \cdot 3 \cdot (2^3 - 1)) = 42$. На основі розробленого програмного забезпечення експериментально встановлено, що мінімальне значення модуля, для якого можна отримати різні синдроми, становить $P = 73$.

На основі формули (3.6) обчислюємо синдром

$$\delta_i = |x'_{k+1} - x_{k+1}|_P = |54 \cdot x'_1 + 25 \cdot x_2 + 67 \cdot x_3|_{73} - |54 \cdot x_1 + 25 \cdot x_2 + 67 \cdot x_3|_{73},$$

і, відповідно, синдром для кожного блоку даних:

$$\delta_{1i} = 54 \cdot e_{x_i} \pmod{73};$$

$$\delta_{2i} = 25 \cdot e_{x_i} \pmod{73};$$

$$\delta_{3i} = 67 \cdot e_{x_i} \pmod{73}.$$

Значення синдрому при виникненні помилки в одному блоці даних наведено в таблиці 3.2.

Таблиця 3.2 – Значення синдрому для всіх можливих варіантів помилок

Значення помилки e_i	Значення синдрому δ_{1i}	Значення синдрому δ_{2i}	Значення синдрому δ_{3i}
1	54	25	67
2	35	50	61
3	16	2	55
4	70	27	49
5	51	52	43
6	32	4	37
7	13	29	31
-1	19	48	6
-2	38	23	12
-3	57	71	18
-4	3	46	24
-5	22	21	30
-6	41	69	36
-7	60	44	42

Як видно із таблиці 3.2, значення синдрому в трьох стовпцях є різними, що дозволяє виявити та виправити помилку в будь-якому одному блоці даних.

Таким чином, запропоновані коректуючі коди характеризуються можливістю гнучкого вибору кількості та розрядності інформаційних символів та швидким виконанням алгоритмів кодування за рахунок паралельного виконання операцій модулярного множення. На основі запропонованих модулярних коректуючих кодів в розділі 5 реалізовано пристрій завадостійкого кодування даних для виправлення однократних помилок.

3.2 Виправлення багатократних помилок на основі модулярних коректуючих кодів

Для підвищення коректуючих можливостей запропонованих модулярних кодів розроблено метод виявлення та виправлення багатократних помилок [72, 87, 93, 219].

Нехай $X \equiv (x_1, x_2, \dots, x_i, \dots, x_k)$ – кортеж символів заданої розрядності, який необхідно передати; $E_i \equiv (0, 0, \dots, e_i, \dots, 0)$ – код помилки; $X' \equiv (x'_1, x'_2, \dots, x'_i, \dots, x'_k)$ – кортеж символів в результаті дії завад:

$$\begin{aligned} (x'_1, x'_2, \dots, x'_i, \dots, x'_k) &= (x_1, x_2, \dots, x_i, \dots, x_k) + \\ &+ (0, 0, \dots, e_i, \dots, 0) = (x_1, |x_2 + e_i|_P, \dots, x_i, \dots, x_k). \end{aligned}$$

Значення контрольного символу

$$x_{k+1} = |(v_1 \cdot x_1 + v_2 \cdot x_2 + \dots + v_i \cdot x_i + \dots + v_k \cdot x_k)|_P, \quad (3.8)$$

де v_i – коефіцієнти, взаємно прості з P ;

$|\bullet|_P$ – операція отримання залишку за модулем P .

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

$$x'_{k+1} = \left| (v_1 \cdot x_1 + v_2 \cdot x_2 + \dots + v_i \cdot x'_i + \dots + v_k \cdot x_k) \right|_P. \quad (3.9)$$

Для виявлення помилки обчислимо синдром δ , який представляє різницю між отриманим перевірочним символом і перевірочним символом, обчисленим на приймальній стороні (в декодері):

$$\delta = \left| x'_{k+1} - x_{k+1} \right|_P.$$

Очевидно, що, якщо синдром дорівнює нулю $\delta = 0$, то помилки немає.

При відсутності помилки $x'_i = x_i$ і, відповідно, $x'_{k+1} = x_{k+1}$. Тобто, якщо $\delta \neq 0$ – є помилка, отже $x'_i \neq x_i$, то, як наслідок, $x'_{k+1} \neq x_{k+1}$.

Для виправлення помилок необхідно, щоб значення синдрому δ було унікальне для всіх можливих варіантів помилки. Дана умова виконується при дотриманні наступних правил:

- 1) значення контрольного модуля $P > 2 \cdot k \cdot (2^m - 1)$, де k – кількість інформаційних символів;
- 2) коефіцієнти v_i – взаємно прості з P , тобто найбільший спільний дільник $\text{gcd}(v_i, P) = 1$.

Локалізація помилки здійснюється шляхом порівняння синдрому δ з таблицею попередньо знайдених розв'язків рівняння $S_{ij} = \left| v_i \cdot e_{ij} \right|_P$, для всіх $-2^m - 1 < e_{ij} < 2^m - 1$, де $i = 1, \dots, n$, $j = -2^m - 1, \dots, 2^m - 1$.

На основі аналізу значень S_{ij} визначаємо символ, в якому відбулася помилка.

Для виправлення помилки в одному символі необхідно розв'язати рівняння

$$\delta = |x'_{k+1} - x_{k+1}|_P = |v_i \cdot |x'_i - x_i|_P|_P.$$

Оскільки $x'_i = |x_i + e_j|_P$,

то

$$|v_i \cdot ||x_i + e_j|_P - x_i|_P|_P = S_{ij},$$

$$|v_i \cdot x_i + v_i \cdot e_j - v_i \cdot x_i|_P = S_{ij},$$

звідси слідує

$$S_{ij} = |v_i \cdot e_j|_P. \quad (3.10)$$

З рівняння (3.10) знаходимо значення помилки e_j .

Відповідно, правильне значення інформаційного символу дорівнює:

$$x_i = x'_i - e_j.$$

Виявлення та виправлення двократних помилок. Для забезпечення можливості виправлення помилок в двох інформаційних символах вводимо додатковий перевірючий символ x_{k+2} і вибираємо взаємно прості коефіцієнти w_i .

Значення перевірючого символу

$$x_{k+2} = |(w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_i \cdot x_i + \dots + w_k \cdot x_k)|_P, \quad (3.11)$$

а з врахуванням помилки

$$x'_{k+2} = \left| (w_1 \cdot x_1 + v_2 \cdot x_2 + \dots + w_i \cdot x'_i + \dots + w_k \cdot x_k) \right|_P.$$

В результаті отримаємо систему рівнянь

$$\begin{cases} \left| (v_1 \cdot x_1 + v_2 \cdot x_2 + \dots + v_i \cdot x'_i + \dots + v_k \cdot x_k) \right|_P = x_{k+1} \\ \left| (w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_i \cdot x'_i + \dots + w_k \cdot x_k) \right|_P = x_{k+2} \end{cases}$$

і визначимо синдром

$$\delta_1 = \left| x'_{k+1} - x_{k+1} \right|_P,$$

$$\delta_2 = \left| x'_{k+2} - x_{k+2} \right|_P,$$

або

$$\delta_1 = \left| v_1 \cdot (x'_1 - x_1) + v_2 \cdot (x'_2 - x_2) + \dots + v_i \cdot (x'_i - x_i) + \dots + v_k \cdot (x'_k - x_k) \right|_P, \quad (3.12)$$

$$\delta_2 = \left| w_1 \cdot (x'_1 - x_1) + w_2 \cdot (x'_2 - x_2) + \dots + w_i \cdot (x'_i - x_i) + \dots + w_k \cdot (x'_k - x_k) \right|_P. \quad (3.13)$$

При $\delta_1 = 0$ і $\delta_2 = 0$ – помилки відсутні, в іншому випадку наявні помилки.

Припустивши, що помилки відбулися в двох символах (за відсутності помилки $x'_i - x_i = 0$), рівняння (3.12) і (3.13) набудуть вигляду:

$$\left| v_i \cdot (x'_i - x_i) + v_{i+1} \cdot (x'_{i+1} - x_{i+1}) \right|_P = \delta_1, \quad (3.14)$$

$$\left| w_i \cdot (x'_i - x_i) + w_{i+1} \cdot (x'_{i+1} - x_{i+1}) \right|_P = \delta_2. \quad (3.15)$$

Розв'язавши рівняння (3.14) і (3.15), отримаємо правильні значення інформаційних символів x_i і x_{i+1} .

На основі наведеного вище можна сформулювати алгоритм локалізації помилки, як послідовність наступних кроків.

1. Обчислити перевірочні символи по прийнятих інформаційних символах x_{k+1} , x_{k+2} .

2. Обчислити синдром δ_1 , δ_2 .

3. Якщо $\delta_1 = 0$, $\delta_2 = 0$ – помилки немає, зупинка і вихід. В іншому випадку продовження.

4. Припускаємо, що помилка відбулася в символах x_i і x_{i+1} , $i = 1, \dots, n$. Збільшуємо i , $i = i + 1$.

5. Розв'язуємо рівняння (3.14) і (3.15).

6. Якщо знайдені $x_i < 2^m - 1$ і $x_{i+1} < 2^m - 1$. Перехід до пункту 2.

7. Збільшуємо i , $i = i + 1$. Перехід до пункту 4.

Для виявлення двократної помилки r в кортежі даних, який складається із k інформаційних символів, необхідно здійснити

$C_k^r = \frac{k!}{r!(k-r)!}$ ітерацій, наприклад, для $k = 8$, $r = 2$, $C_8^2 = 28$ ітерацій.

Розроблений метод та алгоритм виправлення двократних помилок може бути застосований для виправлення багатократних помилок.

Особливістю розроблених коректуючих кодів є можливість зміни кількості перевірочних символів x_{k+i} , $i = 1, \dots, h$, де h - максимальна кількість перевірочних символів без зміни принципів роботи. Тобто, при збільшенні ймовірності помилки в каналі зв'язку достатньо збільшити кількість перевірочних символів. Така можливість дозволяє забезпечити високу надійність передачі даних при зміні характеристик каналу зв'язку.

Можливість виправлення двох помилок проілюструємо на наступному прикладі. Розглянемо модулярний коректуючий код, який складається з восьми інформаційних і двох перевірочних символів, і 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 модуль $P = 1021$ та взаємно прості коефіцієнти для обчислення першого перевірного символу: $v_1 = 13, v_2 = 17, v_3 = 19, v_4 = 23, v_5 = 29, v_6 = 31, v_7 = 37, v_8 = 43$; взаємно прості коефіцієнти для обчислення другого перевірного символу: $w_1 = 7, w_2 = 61, w_3 = 73, w_4 = 83, w_5 = 103, w_6 = 199, w_7 = 239, w_8 = 313$ [102].

Значення перевірочних символів знаходимо за формулою (3.8):

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

$$x_{k+2} = |7 \cdot 5 + 61 \cdot 8 + 73 \cdot 10 + 83 \cdot 3 + 103 \cdot 7 + 199 \cdot 14 + 239 \cdot 12 + 313 \cdot 1|_{1021} = 22.$$

Нехай прийняли дані $X' = (10, 7, 10, 3, 7, 14, 12, 1)$. Обчислимо перевірочні символи по прийнятих даних за формулою (3.9):

$$x'_{k+1} = |13 \cdot 10 + 17 \cdot 7 + 19 \cdot 10 + 23 \cdot 3 + 29 \cdot 7 + 31 \cdot 14 + 37 \cdot 12 + 43 \cdot 1|_{1021} = 611,$$

$$x'_{k+2} = |7 \cdot 10 + 61 \cdot 7 + 73 \cdot 10 + 83 \cdot 3 + 103 \cdot 7 + 199 \cdot 14 + 239 \cdot 12 + 313 \cdot 1|_{1021} = 1017$$

Обчислимо синдром:

$$\delta_1 = |x'_{k+1} - x_{k+1}|_P = |611 - 563|_{1021} = 48.$$

$$\delta_2 = |x'_{k+2} - x_{k+2}|_P = |1017 - 22|_{1021} = 995.$$

Оскільки жодне із значень синдрому не дорівнює нулю, це свідчить про наявність помилки.

Припустивши, що помилка є в символах x_1 і x_2 , рівняння (3.12) і (3.13) набудуть вигляду:

$$|v_1 \cdot e_1 + v_2 \cdot e_2|_P = \delta_1,$$

$$|w_1 \cdot e_1 + w_2 \cdot e_2|_P = \delta_2,$$

де e_1, e_2 – різниця між прийнятим і переданим значенням інформаційного символу:

$$e_1 = |x'_1 - x_1|_P, \quad e_2 = |x'_2 - x_2|_P.$$

Запишемо рівняння (3.12, 3.13) з врахуванням числових значень:

$$|13 \cdot e_1 + 17 \cdot e_2|_P = 48, \quad (3.16)$$

$$|7 \cdot e_1 + 61 \cdot e_2|_P = 995. \quad (3.17)$$

Помноживши рівняння (3.16) на 7, а рівняння (3.17) на 13, отримаємо:

$$|91 \cdot e_1 + 119 \cdot e_2|_P = 336,$$

$$|91 \cdot e_1 + 793 \cdot e_2|_P = 683,$$

Обчисливши різницю цих рівнянь, отримаємо:

$$|347 \cdot e_2|_P = 674.$$

Для розв'язку даного рівняння знайдемо обернений елемент до 347 за модулем 1021, тобто $347^{-1}(\text{mod}1021)$. Помноживши ліву і праву частини на обернений елемент, $347^{-1} = 868$ одержимо:

$$|868 \cdot 347 \cdot e_2|_P = |674 \cdot 868|_P,$$

в результаті отримаємо $e_2 = 1020$.

Для знаходження e_1 підставимо значення e_2 в рівняння (3.17):

$$|7 \cdot e_1 + 61 \cdot 1020|_P = 995,$$

$$|7 \cdot e_1|_P = 35,$$

отже $e_1 = 5$.

Отже, правильні значення інформаційних символів

$$x_1 = |x'_1 - 5|_{1021} = |10 - 5|_{1021} = 5,$$

$$x_2 = |x'_2 - 1020|_{1021} = |7 - 1020|_{1021} = 8.$$

В таблиці 3.3 наведено результати обчислення значень інформаційних символів при всіх можливих варіантах двох помилок в повідомленні.

Як видно з таблиці 3.3, значення інформаційних символів в межах робочого діапазону знаходяться тільки в першому рядку, що відповідає позиції помилок (u_1, u_2) в першому і другому інформаційних символах.

При мінімальній кодовій відстані $d_{\min} \geq 2 \cdot t + 1$ – даний код може виправляти всі сукупності із t або меншої кількості помилок. При введенні двох контрольних символів цей код забезпечить гарантоване виправлення помилок в двох блоках даних.

Таблиця 3.3 – Виявлення двократних помилок

№	Позиція помилки		Значення інформаційних символів	
	u_1	u_2	x_1	x_2
1	1	2	5	8
2	1	3	413	699
3	1	4	223	502
4	1	5	431	732
5	1	6	22	732
6	1	7	405	479
7	1	8	1011	742
8	2	3	531	76
9	2	4	200	169
10	2	5	437	387
11	2	6	308	45
12	2	7	276	439
13	2	8	689	775
14	3	4	135	164
15	3	5	186	735
16	3	6	626	96
17	3	7	577	437
18	3	8	786	678
19	4	5	959	585
20	4	6	598	592
21	4	7	295	795
22	4	8	151	823
23	5	6	281	843
24	5	7	577	309

Продовження таблиці 3.3

25	5	8	1003	1014
26	6	7	232	104
27	6	8	718	917
28	7	8	994	817

При необхідності використання двох і більше контрольних символів існують наступні варіанти вибору коефіцієнтів лінійної форми [152]:

- 1) однакові контрольні модулі і різні коефіцієнти лінійної форми;
- 2) різні модулі і однакові коефіцієнти лінійної форми.

Розроблені коректуючі коди на основі модулярної арифметики дозволяють ефективно виправляти багатократні помилки в пакетах даних. Дані коди забезпечують можливість адаптивно підвищувати надійність передачі даних в безпроводних сенсорних мережах за рахунок збільшення кількості перевірочних символів.

3.2.1 Експериментальні дослідження модулярних коректуючих кодів

Коректуючий код на основі модулярної арифметики використано для підвищення надійності передачі даних в безпроводних сенсорних мережах систем промислової автоматизації.

В [160] показано, що перевірочні символи всіх блоків доцільно зберігати в кінці пакету. Таке розміщення перевірочних символів дозволить приймати пакети даних безпроводним сенсорам, які не використовують контроль помилок на основі коректуючих кодів. Тобто, забезпечить сумісність з мережами стандарту IEEE 802.15.4.

Для використання розробленого модулярного коректуючого коду на MAC рівні протоколу стандарту IEEE 802.15.4 розділимо дані (поля MHR і MAC payload) на п'ять блоків (D1 – D5) (рисунок 3.1) [157] з довжиною

інформаційного блоку $k_i = 17$ байт. Перевірочні блоки R1 – R5 формуються за відповідним блоком даних і зберігаються після останнього блоку даних.

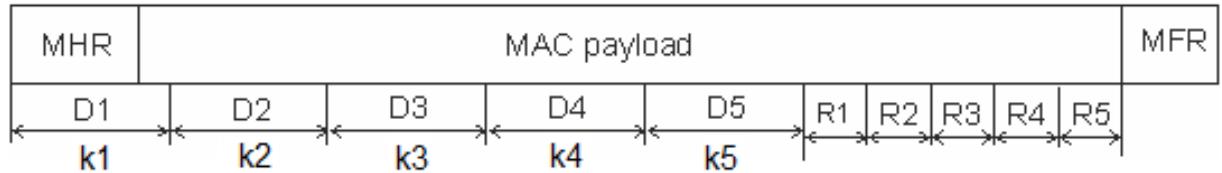


Рисунок 3.1 – Структура фрейму даних з використанням коректуючих кодів

При отриманні даних здійснюється перевірка цілісності пакету з використанням CRC кодів. Якщо помилки в пакеті не виявлено, то додаткова перевірка з використанням модулярного коду не виконується. При виявленні в пакеті помилок за допомогою CRC кодів здійснюється виправлення помилок з використанням запропонованих модулярних коректуючих кодів.

Експериментальні дослідження часу формування коректуючого коду проводилися в середовищі Matlab для кодів Ріда-Соломона RS (127, 87), коректуючого коду СЗК та модулярного коректуючого коду (М - код) з параметрами $n = 25$, $k = 17$, $r = 8$, $t = 4$. Результати дослідження часу формування коректуючи символів наведено в таблиці 3.4.

Таблиця 3.4 – Параметри коректуючих кодів

Код	Час формування коду, мс	Швидкість коду, $R = \frac{k}{n}$
Код СЗК	117,5	0,64
М – код	22,5	0,68
RS (127, 87)	109,7	0,685

Як видно з таблиці 3.4, модулярний коректуючий код (М – код) дозволяє приблизно в 5 разів зменшити час формування коду порівняно з

коректуючим кодом СЗК та кодом RS (127, 87) при майже однаковій швидкості коду.

На основі формули (3.7) розраховано теоретичну та експериментальну швидкість коду (рисунок 3.2). Теоретична швидкість коду відповідає мінімально можливому значенню модуля для заданої кількості інформаційних символів. Експериментальна швидкість коду обчислена з використанням реальних значень модулів, які знайдені на основі розробленого програмного забезпечення. При збільшенні кількості інформаційних символів у повідомленні швидкість коду зростає (рисунок 3.2).

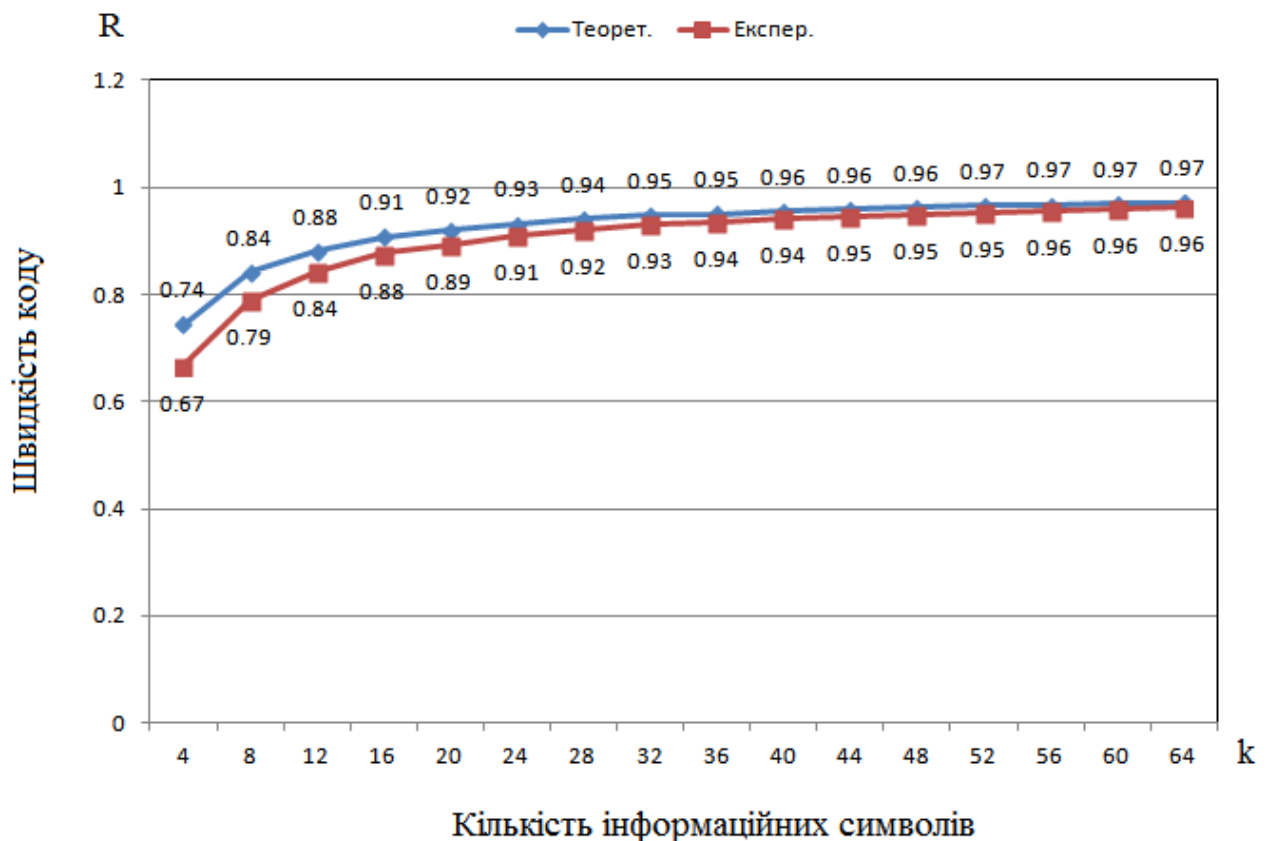


Рисунок 3.2 – Залежність швидкості коду від кількості інформаційних символів

Проведено порівняння апаратної складності процесу завадостійкого кодування даних при реалізації на ПЛІС коректуючих кодів СЗК (код СЗК) та розроблених модулярних коректуючих кодів з наступними параметрами:

розрядність вхідних символів – 8 біт; кількість інформаційних символів – 2–8 біт (рисунок 3.3). Як видно з рисунку 3.3, апаратна складність процесу кодування модулярних коректуючих кодів в середньому в 16 разів менша за коректуючі коди СЗК.

На рисунку 3.4 наведено результати дослідження часу затримки формування вихідних сигналів при кодуванні даних з використанням коректуючих кодів СЗК та модулярних коректуючих кодів. Проведені експериментальні дослідження показали, що затримка часу формування вихідних сигналів при реалізації на ПЛІС в модулярних коректуючих кодах приблизно в 2,2 рази менша порівняно з коректуючими кодами СЗК (рисунок 3.4).

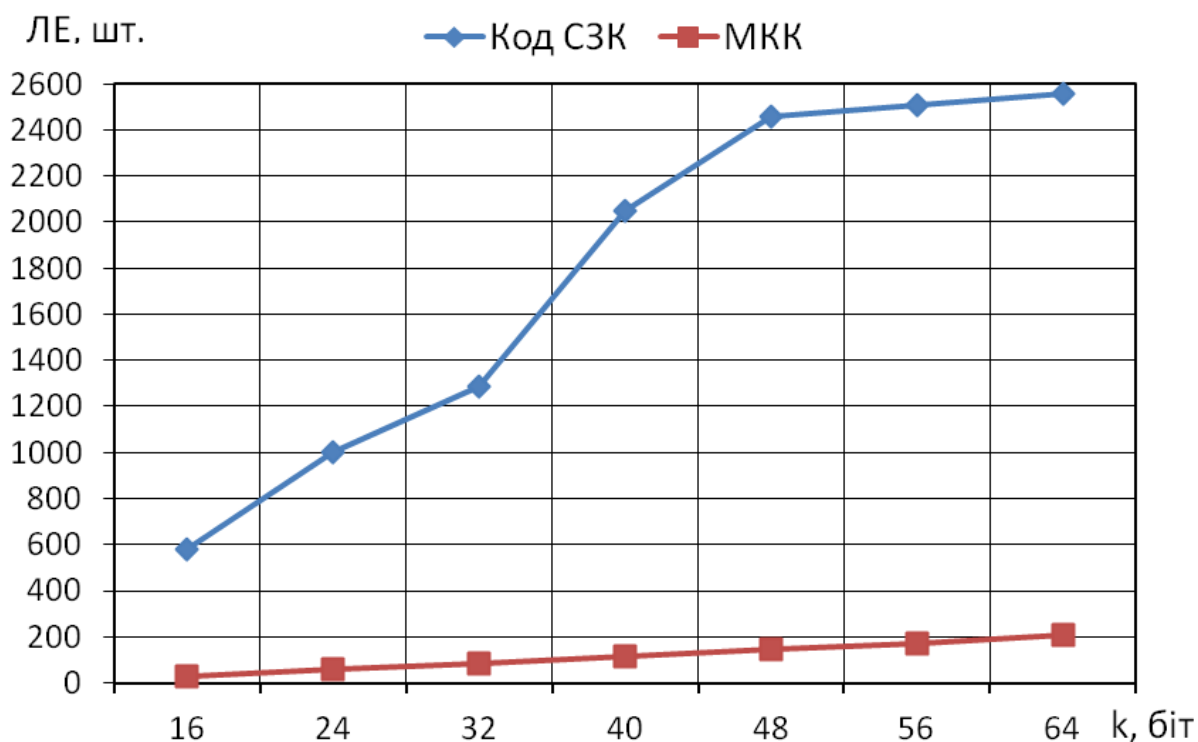


Рисунок 3.3 – Залежність апаратних затрат від розрядності вхідних даних: ЛЕ – логічні елементи, k – розрядність вхідних даних.

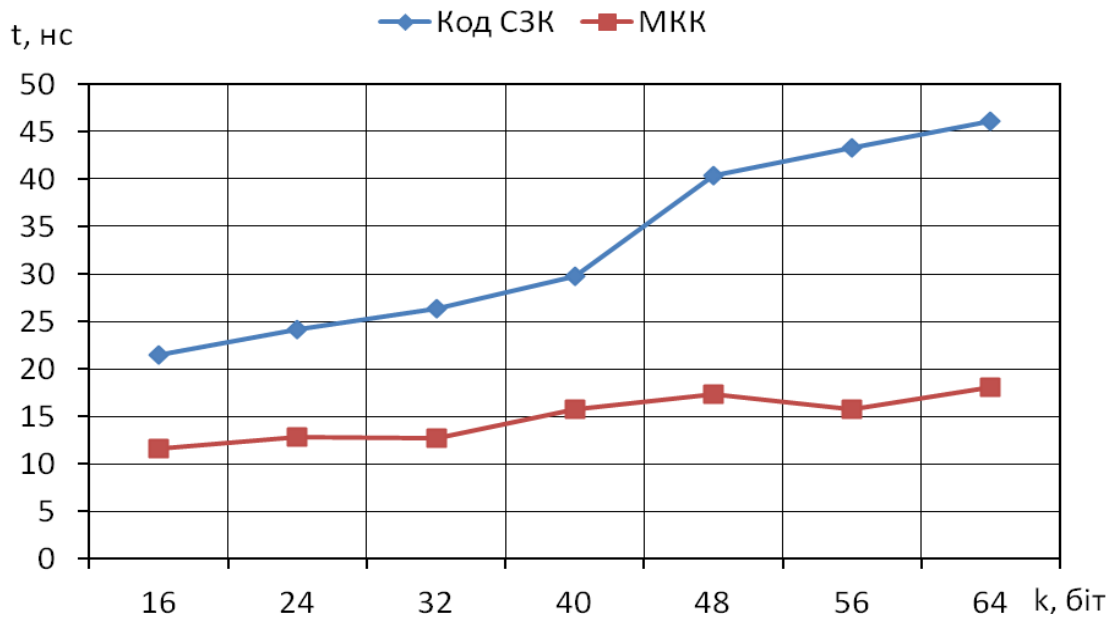


Рисунок 3.4 – Залежність часу формування вихідних сигналів від розрядності вхідних даних.

Ефективність кодування розраховано за формулою (1.3) та проведено в пакеті Communications Toolbox 3.0 системи Matlab з використанням графічного середовища BERTool. Графічні залежності (рисунок 3.5) отримано для модуляції Offset Quadrature Phase-shift keying (OQPSK), яка використовується в стандарті IEEE 802.15.4, та розробленого модулярного коректуючого коду з мінімальною кодовою відстанню $d_{\min} = 9$.

Як видно з рисунку 3.5 (точки А і В), використання завадостійкого кодування дозволяє зменшити E_b/N_0 з 10,5 до 7,5 дБ при збереженні заданої надійності передачі, тобто ефективність кодування дорівнює 3 дБ.

Зазначимо, що використання в БСМ коректуючих кодів, які виявляють та виправляють помилки, дозволить підвищити загальну пропускну здатність за рахунок зменшення кількості повторних передач. В свою чергу, розроблений модулярний коректуючий код зменшує відношення E_b/N_0 на 3 дБ при збереженні заданої достовірності передачі даних.

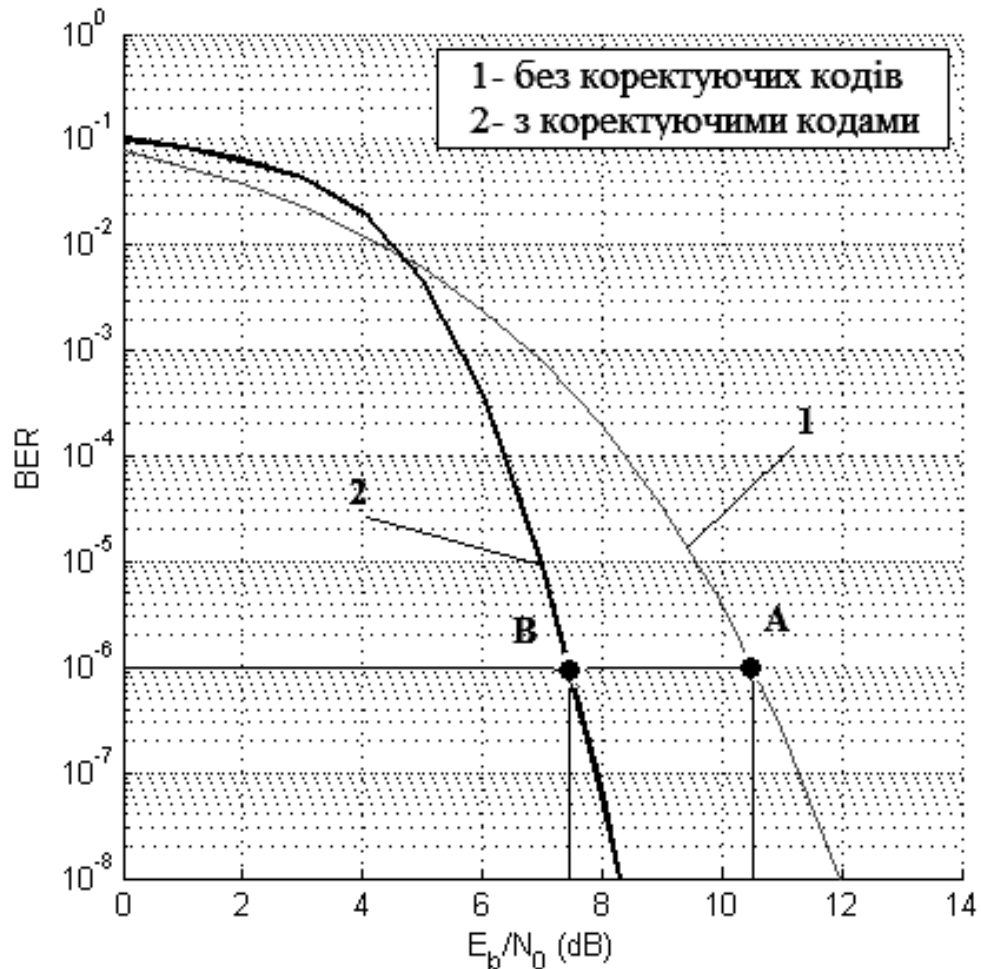


Рисунок 3.5 – Порівняння надійності передачі даних при використанні схеми без кодування і з кодуванням: BER – ймовірність бітової помилки.

Таким чином, проведені дослідження часу формування коду показали, що модулярний коректуючий код зменшує час формування коду приблизно в 5 разів, порівняно з R – кодом СЗК та кодом Ріда – Соломона RS (127, 87) при практично однаковій швидкості коду. При цьому апаратні затрати на реалізацію кодування в середньому в 16 разів менші, порівняно з коректуючими кодами системи залишкових класів. Розміщення перевірочних символів в кінці поля даних протоколу забезпечує сумісність з мережами стандарту IEEE 802.15.4.

3.3 Двовимірний контроль помилок на основі модулярних коректуючих кодів

Для виправлення пакетів помилок в повідомленні запропоновано двовимірні коректуючі коди, які базуються на модулярних коректуючих кодах, але на відміну від останніх забезпечують двовимірний контроль помилок, що підвищує їх коректуючі можливості [88, 194, 225]. Під двовимірним контролем в роботі розуміється формування перевірочних символів по рядках та стовпцях матриці даних.

Перевірочний символ по рядку:

$$x_{i,k+1} = (v_1 \cdot x_{i,1} + v_2 \cdot x_{i,2} + \dots + v_i \cdot x_{i,i} + \dots + v_k \cdot x_{i,k}) \pmod{p}, \quad (3.18)$$

де p – контрольний модуль;

v_i – взаємно прості числа з p ;

x_i – інформаційні символи заданої розрядності, подані в позиційній систем числення.

Значення контрольного модулю вибираємо за формулою (3.7):

Перевірочний символ по стовпцю:

$$x_{k+2,j} = (v_1 \cdot x_{1,j} + v_2 \cdot x_{2,j} + \dots + v_i \cdot x_{i,j} + \dots + v_k \cdot x_{k,j}) \pmod{p}. \quad (3.19)$$

Двовимірний контроль помилок представимо у вигляді матриці:

$$\begin{array}{cccccc|c}
 x_{1,1} & x_{1,2} & \dots & x_{1,j} & \dots & x_{1,k} & x_{1,k+1} \\
 x_{2,1} & x_{2,2} & \dots & x_{2,j} & \dots & x_{2,k} & x_{2,k+1} \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 x_{i,1} & x_{i,2} & \dots & x_{i,j} & \dots & x_{i,k} & x_{i,k+1} \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 x_{k,1} & x_{k,2} & \dots & x_{k,j} & \dots & x_{k,k} & x_{k,k+1} \\
 \hline
 x_{k+2,1} & x_{k+2,2} & \dots & x_{k+2,j} & \dots & x_{k+2,k} &
 \end{array}$$

На приймальній стороні декодер за формулами (3.18), (3.19) обчислює перевірочні символи $x'_{i,k+1}$ і $x'_{k+2,j}$ по отриманих даних. В результаті обчислення отримаємо матрицю:

$$\begin{array}{cccccc|c}
 x_{1,1} & x_{1,2} & \dots & x'_{1,j} & \dots & x_{1,k} & x'_{1,k+1} \\
 x_{2,1} & x_{2,2} & \dots & x'_{2,j} & \dots & x_{2,k} & x'_{2,k+1} \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 x_{i,1} & x_{i,2} & \dots & x'_{i,j} & \dots & x_{i,k} & x'_{i,k+1} \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 x_{k,1} & x_{k,2} & \dots & x'_{k,j} & \dots & x_{k,k} & x'_{n,k+1} \\
 \hline
 x'_{k+2,1} & x'_{k+2,2} & \dots & x'_{k+2,j} & \dots & x'_{k+2,k} &
 \end{array}$$

Виявлення помилок відбувається на основі аналізу синдрому

$$\delta_{i,j} = (x'_{i,j} - x_{i,j}) \bmod p, \quad (3.20)$$

або

$$\delta_i = \begin{array}{|l}
 (x'_{1,k+1} - x_{1,k+1}) \bmod p \\
 (x'_{2,k+1} - x_{2,k+1}) \bmod p \\
 \dots \\
 (x'_{i,k+1} - x_{i,k+1}) \bmod p \\
 \dots \\
 (x'_{k,k+1} - x_{k,k+1}) \bmod p
 \end{array}, \quad \delta_j = \begin{array}{|l}
 (x'_{k+2,1} - x_{k+2,1}) \bmod p \\
 (x'_{k+2,2} - x_{k+2,2}) \bmod p \\
 \dots \\
 (x'_{k+2,j} - x_{k+2,j}) \bmod p \\
 \dots \\
 (x'_{k+2,k} - x_{k+2,k}) \bmod p
 \end{array},$$

де $x'_{i,k+1}$ – перевірочний символ, обчислений по прийнятих даних.

Отже, якщо $\delta_{i,j} = 0$ – помилки немає, відповідно синдром

$$\delta_i = \begin{vmatrix} 0 \\ 0 \\ \dots \\ 0 \\ \dots \\ 0 \end{vmatrix} \quad \text{і} \quad \delta_j = \begin{vmatrix} 0 \\ 0 \\ \dots \\ 0 \\ \dots \\ 0 \end{vmatrix} \quad \text{– помилки немає.}$$

В іншому випадку при $\delta_{i,j} \neq 0$ на перетині відповідного рядка та стовпця виникла помилка. Отже, ненульове значення синдрому однозначно вказує на спотворений інформаційний символ.

При двох і більше спотворених інформаційних символах в одному рядку (при умові, що одна помилка в стовпці) виправлення помилок відбувається на основі аналізу синдрому відповідного стовпця.

Аналогічно, якщо спотворено два і більше інформаційних символи, які розміщені в одному стовпці, то виправлення помилок реалізується на основі аналізу синдрому відповідного рядка.

Для виправлення помилок в інформаційних символах, розміщених в $n-1$ рядку і $n-1$ стовпцю, необхідно розв'язати рівняння:

$$(v_1 \cdot (x'_{i,1} - x_{i,1}) + v_2 \cdot (x'_{i,2} - x_{i,2}) + \dots + v_i \cdot (x'_{i,j} - x_{i,j}) + \dots + v_n \cdot (x'_{i,k} - x_{i,k})) \bmod p = \delta_{i,1} \quad (3.21)$$

Якщо найбільший спільний дільник $\gcd(v_i, p) = 1$, тобто (v_i, p) – взаємно прості числа, то рівняння (3.22) має єдиний розв'язок, який буде відповідати правильним інформаційним символам.

Експериментальні дослідження методу. Розглянемо приклад передачі даних, які сформовані у вигляді матриці розміром $k \times k$, при $k = 8$, $m = 8$, де k – розмірність матриці даних, m – розрядність інформаційних символів:

$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	$x_{1,5}$	$x_{1,6}$	$x_{1,7}$	$x_{1,8}$	$x_{1,9}$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	$x_{2,5}$	$x_{2,6}$	$x_{2,7}$	$x_{2,8}$	$x_{2,9}$
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$	$x_{3,5}$	$x_{3,6}$	$x_{3,7}$	$x_{3,8}$	$x_{3,9}$
$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$	$x_{4,5}$	$x_{4,6}$	$x_{4,7}$	$x_{4,8}$	$x_{4,9}$
$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	$x_{5,4}$	$x_{5,5}$	$x_{5,6}$	$x_{5,7}$	$x_{5,8}$	$x_{5,9}$
$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	$x_{6,4}$	$x_{6,5}$	$x_{6,6}$	$x_{6,7}$	$x_{6,8}$	$x_{6,9}$
$x_{7,1}$	$x_{7,2}$	$x_{7,3}$	$x_{7,4}$	$x_{7,5}$	$x_{7,6}$	$x_{7,7}$	$x_{7,8}$	$x_{7,9}$
$x_{8,1}$	$x_{8,2}$	$x_{8,3}$	$x_{8,4}$	$x_{8,5}$	$x_{8,6}$	$x_{8,7}$	$x_{8,8}$	$x_{8,9}$
$x_{9,1}$	$x_{9,2}$	$x_{9,3}$	$x_{9,4}$	$x_{9,5}$	$x_{9,6}$	$x_{9,7}$	$x_{9,8}$	

За допомогою розробленого програмного забезпечення експериментально знайдено значення взаємно простих коефіцієнтів v_i та контрольного модуля p : $p = 131071$, $v_1 = 3$, $v_2 = 257$, $v_3 = 263$, $v_4 = 269$, $v_5 = 271$, $v_6 = 277$, $v_7 = 281$, $v_8 = 283$.

В таблиці 3.5 наведено значення синдромів

$$\delta_{i,j} = v_j \cdot e_i \pmod{p}, \quad 1 \leq j \leq 8,$$

розрахованих для всіх можливих варіантів помилок x_i .

Для інформаційних символів розрядністю 8 біт значення помилки знаходиться в діапазоні $-255 \leq e_i \leq 255$.

Таблиця 3.5 – Значення синдрому для восьми інформаційних символів

Значення помилки e_i	Значення синдрому δ_{i1}	Значення синдрому δ_{i2}	Значення синдрому δ_{i3}	Значення синдрому δ_{i4}	Значення синдрому δ_{i5}	Значення синдрому δ_{i6}	Значення синдрому δ_{i7}	Значення синдрому δ_{i8}
1	211	257	263	269	271	277	281	283
2	422	514	526	538	542	554	562	566
3	633	771	789	807	813	831	843	849
4	844	1028	1052	1076	1084	1108	1124	1132
...			
255	53805	65535	67065	68595	69105	70635	71655	72165
-1	130860	130814	130808	130802	130800	130794	130790	130788
-2	130649	130557	130545	130533	130529	130517	130509	130505
-3	130438	130300	130282	130264	130258	130240	130228	130222
-4	130227	130043	130019	129995	129987	129963	129947	129939
...			
-255	77266	65536	64006	62476	61966	60436	59416	58906

На основі запропонованого методу розроблено алгоритм кодування, що складається з наступних кроків.

1. З даних, які підлягають передачі, формується матриця розміру $k \times k$.
2. Вибирається перевірочний модуль p та взаємно прості коефіцієнти $v_1, \dots, v_i, \dots, v_k$.
3. За формулою (3.18) обчислюються перевірочні символи по рядках матриці.
4. За формулою (3.19) обчислюються перевірочні символи по стовпцях матриці.

Також розроблено алгоритм виявлення помилок, що складається з

наступних кроків.

1. З отриманих даних формується матриця розміру $k \times k$.
2. За формулою (3.18) обчислюються перевірочні символи по рядках матриці.
3. За формулою (3.19) обчислюються перевірочні символи по стовпцях матриці.
4. Обчислюється синдром $\delta_{i,j} = (x'_{i,j} - x_{i,j}) \bmod p$ по рядках і стовпцях матриці даних.
5. При $\delta_{i,j} = 0$, відповідно $x_{i,j} = x'_{i,j}$, отже, помилки немає.
6. Якщо $\delta_{i,j} \neq 0$, помилка виявлена.

При спотворенні одного інформаційного символу значення синдрому на перетині рядка і стовпця не дорівнює нулю.

Розглянемо можливі випадки спотворення інформаційних символів.

1. Синдром не дорівнює нулю тільки в одному рядку і стовпці. Спотворення зазнав символ на перетині відповідного рядка і стовпця. Виправлення помилок відбувається по таблиці синдромів (таблиця 3.5).

2. Значення синдрому $\delta_{i,j} \neq 0$ в двох і більше рядках і одному стовпці. Спотворення зазнали інформаційні символи, які знаходяться в одному стовпці. Виправлення помилок відбувається по таблиці синдромів (таблиця 3.5).

3. Значення синдрому $\delta_{i,j} \neq 0$ в двох і більше стовпцях і одному рядку. Спотворення зазнали інформаційні символи, які знаходяться в одному рядку. Виправлення помилок відбувається по таблиці синдромів (таблиця 3.5).

4. Значення синдрому $\delta_{i,j} \neq 0$ в двох і більше рядках і стовпцях. Для виправлення помилок необхідно розв'язати рівняння:

$$(v_1 \cdot (x'_{1,1} - x_{1,1}) + v_2 \cdot (x'_{1,2} - x_{1,2}) + \dots + v_i \cdot (x'_{i,j} - x_{i,j}) + \dots + v_n \cdot (x'_{1,k} - x_{1,k})) \bmod p = \delta_{1,j} \quad (3.22)$$

Нехай спотворені чотири символи, розміщені на перетині двох рядків і двох стовпців. Для початку припустимо, що помилки відбулися в двох символах, розміщених в першому рядку. Оскільки при відсутності помилки $x'_{i,j} - x_{i,j} = 0$ рівняння (3.22) можна записати у вигляді:

$$(v_1 \cdot (x'_{1,1} - x_{1,1}) + v_2 \cdot (x'_{1,2} - x_{1,2})) \bmod p = \delta_{1,1}. \quad (3.23)$$

Оскільки коефіцієнти v_i і p – взаємно прості числа, то рівняння (3.23) має єдиний розв'язок в цілих числах. Зауважимо, що знайдені розв'язки будуть відповідати виправленим значенням інформаційних символів $x_{1,1}$, $x_{1,2}$.

Підставивши $x_{1,1}$, $x_{1,2}$ в рівняння (3.23), перевіряємо значення синдрому $\delta_{i,j}$. При цьому можливі наступні варіанти: 1) синдром першого рядка $\delta_{1,1} = 0$ – помилки виправлені; 2) синдром першого стовпця $\delta_{2,1} \neq 0$. Тоді по таблиці 3.15 знаходимо помилку, яка відповідає синдрому $\delta_{2,1}$, і виправляємо її в наступний спосіб: $x_{2,i} = \delta_{2,1} - e_i$.

У випадку, якщо припущення виявилися невірними, перевіряємо наступну групу символів.

Запропонований двовимірний контроль помилок на основі модулярних коректуючих кодів забезпечує виявлення та виправлення k помилок, які знаходяться в одному рядку або стовпці матриці даних розміру $k \times k, 2 \cdot k$ помилок при умові, що вони знаходяться в двох рядках або двох стовпцях матриці даних, що може бути використано в БСМ, які експлуатуються на підприємствах з високим рівнем електромагнітних завад.

3.4 Виявлення та виправлення помилок при виконанні арифметичних операцій

Забезпечення високої надійності роботи обчислювальних засобів, які функціонують в системах критичного призначення, є актуальною науковою проблемою [15]. Для контролю правильності виконання арифметичних операцій використовують властивості порівняння за модулем, на основі яких побудовані спеціальні арифметичні коди [2, 182]. Однак проблема підвищення надійності функціонування обчислювальних засобів, зокрема виявлення та виправлення помилок при виконанні арифметичних операцій в системах, які функціонують в позиційних системах числення, залишається актуальною.

Розглянемо можливість виявлення та виправлення помилок при виконанні арифметичних операцій на основі розроблених модулярних коректуючих кодів.

Нехай арифметичні операції виконуються над операндами A , B , де $A = (a_{k-1}, \dots, a_i, \dots, a_1, a_0)$, $B = (b_{k-1}, \dots, b_i, \dots, b_1, b_0)$, a_i , b_i – розряди чисел A , B подані в вісімковій або десятковій системах числення.

Суть запропонованого методу виявлення та виправлення помилок при виконанні арифметичних операцій полягає в наступному: для чисел A , B , над якими виконуються арифметичні операції, обчислюються контрольні символи [90]:

$$a_k = |(v_{k-1} \cdot a_{k-1} + \dots + v_i \cdot a_i + \dots + v_0 \cdot a_0)|_p, \quad (3.24)$$

$$b_k = |(v_{k-1} \cdot b_{k-1} + \dots + v_i \cdot b_i + \dots + v_0 \cdot b_0)|_p, \quad (3.25)$$

де p – просте число,

v_i – коефіцієнти взаємно прості з p .

Арифметичні операції додавання, віднімання і множення виконуються відповідно над інформаційними розрядами чисел A , B і над контрольними розрядами. З отриманого результату виконання арифметичних операцій над інформаційними символами обчислюється контрольний символ. Далі порівнюються значення контрольного символу, отриманого в результаті додавання контрольних символів чисел, і контрольного символу, отриманого в результаті додавання інформаційних розрядів [90]:

Операція додавання. Додавання інформаційних розрядів:

$$\begin{array}{r} (a_{k-1}, \dots, a_i, \dots, a_1, a_0) \\ + (b_{k-1}, \dots, b_i, \dots, b_1, b_0) \\ \hline (c_{k-1}, \dots, c_i, \dots, c_1, c_0) \end{array}$$

де c_i – сума чисел a_i , b_i без врахування переносів між розрядами.

Сума контрольних символів:

$$|a_k + b_k|_p = c_k.$$

Обчислюємо контрольний символ від результату додавання інформаційних розрядів:

$$c'_k = |(v_{k-1} \cdot c_{k-1} + \dots + v_i \cdot c_i + \dots + v_0 \cdot c_0)|_p.$$

Для виявлення помилки знайдемо різницю між двома перевірочними символами

$$s = |(c'_k - c_k)|_p.$$

Якщо $s = 0$ – помилки не має, а при $s \neq 0$ – наявна помилка.

Покажемо, що при відсутності помилок $c'_k = c_k$. Для цього запишемо значення відповідно

$$c'_k = |(v_{k-1} \cdot (a_{k-1} + b_{k-1}) + \dots + v_i \cdot (a_i + b_i) + \dots + v_0 \cdot (a_0 + b_0))|_p \quad (3.26)$$

і

$$c_k = |a_k + b_k|_p = \left| \begin{array}{l} (v_{k-1} \cdot (a_{k-1} + b_{k-1}) + \dots \\ + v_i \cdot (a_i + b_i) + \dots + v_0 \cdot (a_0 + b_0)) \end{array} \right|_p. \quad (3.27)$$

Оскільки праві частини виразів (3.26) і (3.27) однакові, то і ліві частини при відсутності помилок при виконанні арифметичних операцій будуть однаковими.

Розглянемо виявлення помилок при виконанні операції *віднімання* двох чисел. Віднімання інформаційних розрядів двох чисел A, B :

$$(a_{k-1}, \dots, a_i, \dots, a_1, a_0) - (b_{k-1}, \dots, b_i, \dots, b_1, b_0) = d_{k-1}, \dots, d_i, \dots, d_1, d_0,$$

де d_i – різниця чисел a_i, b_i за модулем p , $d_i = |a_i - b_i|_p$.

Значення контрольних символів обчислюються за формулами 3.24, 3.25, а різниця контрольних символів

$$|a_k - b_k|_p = d_k.$$

Значення контрольного символу від результату віднімання інформаційних розрядів:

$$d'_k = |(v_{k-1} \cdot d_{k-1} + \dots + v_i \cdot d_i + \dots + v_0 \cdot d_0)|_p.$$

Для виявлення помилки знайдемо різницю між двома перевірочними символами

$$s = |(d'_k - d_k)|_p,$$

аналіз якої показує: якщо $s = 0$ – помилки немає, в іншому випадку наявна помилка.

Аналогічно операції додавання, наведеній вище, можна показати, що при відсутності помилок значення контрольних символів будуть рівними, тобто $d'_k = d_k$.

Операція множення. Для виявлення помилок при виконанні операції множення чисел A , B обчислимо лінійну згортку розрядів двох чисел $(a_{k-1}, \dots, a_i, \dots, a_1, a_0)$ і $(b_{k-1}, \dots, b_i, \dots, b_1, b_0)$ за формулою:

$$m(n) = \sum_{i=0}^n a(i) \cdot b(n-i),$$

де $n = 0 \dots (2 \cdot N - 2)$ при розрядності чисел $N - 1$.

При лінійній згортці розряди чисел зсуваються один відносно іншого, попарно перемножуються і додаються (таблиця 3.6).

Таблиця 3.6 – Згортка трирозрядних чисел A , B .

Розряди чисел A і B						Значення	
		a_0	a_1	a_2			
b_2	b_1	b_0				$m_0 = a_0 \cdot b_0$	
	b_2	b_1	b_0			$m_1 = a_1 \cdot b_0 + a_0 \cdot b_1$	
		b_2	b_1	b_0		$m_3 = a_2 \cdot b_0 + a_1 \cdot b_1 + a_0 \cdot b_2$	
			b_2	b_1	b_0	$m_4 = a_2 \cdot b_1 + a_1 \cdot b_2$	
				b_2	b_1	b_0	$m_5 = a_2 \cdot b_2$

В результаті отримаємо послідовність $(m_{2 \cdot k-1}, \dots, m_i, \dots, m_1, m_0)$, де m_i – розряди послідовності, які представляють добуток чисел A , B без врахування переносів між розрядами. Для отримання правильного результату множення необхідно провести корекцію переносів між розрядами.

Контрольний символ операції множення обчислюємо, як добуток контрольних символів чисел A , B :

$$|a_k \cdot b_k|_p = m_k.$$

Контрольний символ від послідовності $(m_{2 \cdot k-1}, \dots, m_i, \dots, m_1, m_0)$, за результатами згортки

$$m'_{2k} = |(z_{2k-1} \cdot m_{2k-1} + \dots + z_i \cdot m_i + \dots + z_0 \cdot m_0)|_p,$$

де z_i – значення, отримані в результаті згортки взаємно простих коефіцієнтів v_i .

Для виявлення помилки обчислюємо різницю між двома перевірочними символами

$$s = |(m'_{2k} - m_{2k})|_p.$$

Якщо $s = 0$ – помилки немає, а при $s \neq 0$ – наявна помилка.

На основі запропонованого підходу можна сформувати узагальнений алгоритм виявлення помилок при виконанні арифметичних операцій, який складається з наступних кроків.

1. Обчислення контрольних символів операндів A , B ;
2. Виконання арифметичної операції (додавання, віднімання, множення) над операндами A , B .
3. Виконання арифметичної операції (додавання, віднімання, множення) над контрольними символами операндів A , B .

4. Обчислення контрольного символу від отриманого результату виконання заданої арифметичної операції.

5. Виявлення помилки (Порівняння обчислених в пунктах 3 і 4 контрольних символів).

6. виправлення помилок.

При цьому виправлення помилки відбувається за алгоритмом виправлення помилок при передачі даних з використанням модулярних коректуючих кодів (див. пункт 3.1).

На основі розробленого алгоритму, описаного вище, можна синтезувати схему виявлення та виправлення помилок при виконанні арифметичних операцій (рисунок 3.6).

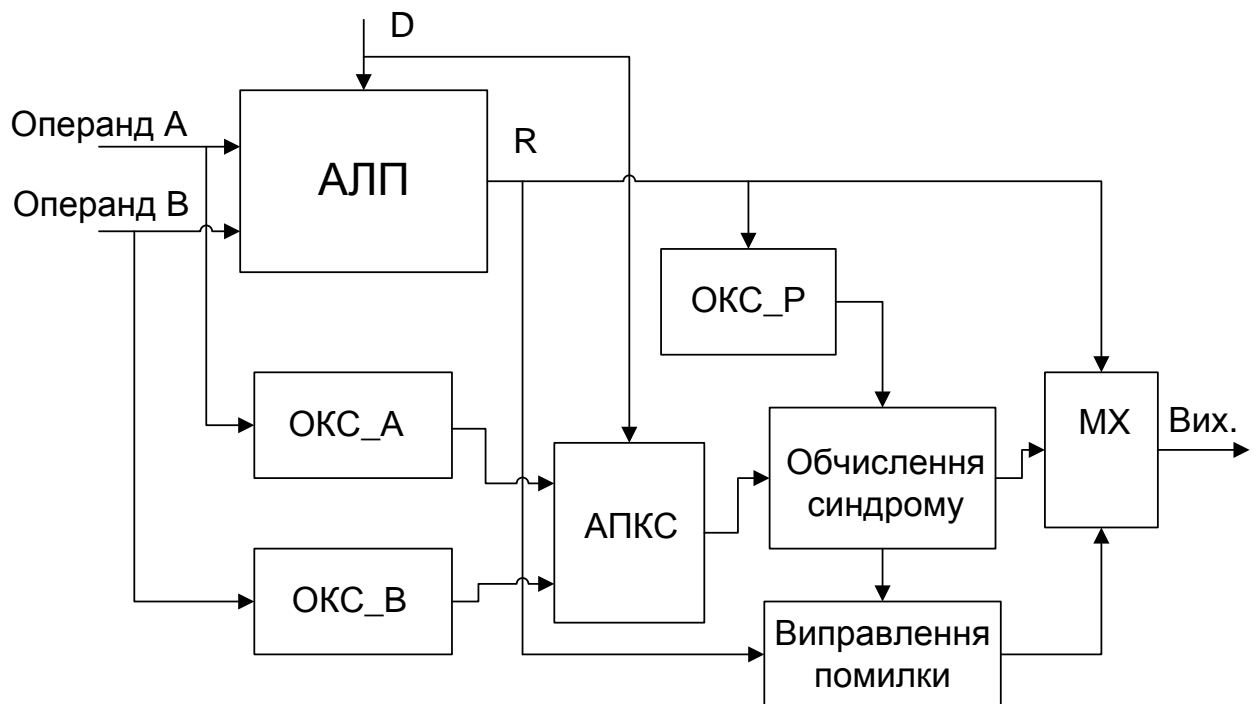


Рисунок 3.6 – Структурна схема блоку виявлення помилок при виконанні арифметичних операцій: АЛП – арифметико логічний пристрій, ОКС_А – обчислювач контрольного символу А, ОКС_В – обчислювач контрольного символу В, ОКС_Р – обчислювач контрольного символу результату, АПКС –

арифметичний пристрій виконання операцій над контрольними символами, MX – мультиплексом, R – результат обчислення операндів, D – код операції.

Операнди A , B поступають на АЛП, а також на ОКС_А і ОКС_В, де відбувається обчислення контрольного символу кожного операнду. Останні подаються на вхід блоку АПКС, де виконується задана арифметична операція над контрольними символами операндів. Результат виконання арифметичної операції R в АЛП поступає на блок ОКС_Р, в якому обчислюється контрольний символ результату виконання арифметичної операції. В блоці обчислення синдрому визначається різниця між контрольним символом результату виконання арифметичної операції і значенням, отриманим в результаті виконання арифметичної операції над контрольними символами операндів. Якщо результат обчислення синдрому дорівнює нулю, то на вихід мультиплексора (MX) поступають дані з АЛП. Якщо ж результат обчислення синдрому не дорівнює нулю, то на вихід мультиплексора поступають дані з блоку виправлення помилок.

Таким чином, в даному підрозділі запропоновано метод виявлення та виправлення помилок при виконанні арифметичних операцій (додавання, віднімання, множення) на основі модулярних коректуючих кодів, який дозволяє виявляти помилки в двох десяткових розрядах та виправляти помилки в одному десятковому розряді при використанні одного контрольного символу.

3.5 Підвищення надійності передачі даних в безпроводних сенсорних мережах на основі декомпозиції

Як показано в розділі 1, при передачі мультимедійних даних в безпроводних сенсорних мережах необхідно враховувати обмеження, що накладаються на пропускну здатність каналів зв'язку та обсяг пам'яті безпроводного модуля.

Мультимедійні дані чутливі до втрати пакетів зокрема втрата невеликої частини даних зображення призводить до відкидання всього зображення або до різкого зниження якості зображення. Враховуючи обмежений розмір поля даних структури протоколів БСМ, мультимедійний контент розділяється на порівняно велику кількість пакетів для відновлення зображення, причому ці пакети не повинні бути втрачені або спотворені в процесі їх передачі.

Існуючі методи стиснення та формати зберігання мультимедійного контенту не є стійкі до впливу завад, які виникають в безпроводних каналах зв'язку і приводять до спотворення даних [113, 116, 177]. Спотворення одного біту стиснутих даних унеможлиблює відновлення всього повідомлення (зображення). Отже, необхідно покращити існуючі методи передачі пакетів, з урахуванням характеристик мультимедійного контенту та типу завад. Для цього запропоновано використати модулярні коректуючі коди та принцип декомпозиції даних [99].

Нехай необхідно передати масив даних $A_1, A_2, \dots, A_k, A_{k+1}$ захищених модулярним коректуючим кодом:

$$A_1 = (x_1^1, x_2^1, \dots, x_i^1, \dots, x_k^1, x_{k+1}^1),$$

$$A_2 = (x_1^2, x_2^2, \dots, x_i^2, \dots, x_k^2, x_{k+1}^2),$$

$$A_j = (x_1^j, x_2^j, \dots, x_i^j, \dots, x_k^j, x_{k+1}^j),$$

$$A_k = (x_1^k, x_2^k, \dots, x_i^k, \dots, x_k^k, x_{k+1}^k),$$

де x_i^j – інформаційний символ розрядності m ;

k – кількість інформаційних символів в пакеті.

Принцип декомпозиції полягає в тому, що після обчислення перевірочних символів x_{k+1}^h нові пакети формуються по стовпцях, тобто в наступній послідовності:

$$\left(x_1^1, x_1^2, \dots, x_1^h\right), \left(x_2^1, x_2^2, \dots, x_2^h\right), \left(x_k^1, x_k^2, \dots, x_k^h\right), \left(x_{k+1}^1, x_{k+1}^2, \dots, x_{k+1}^h\right),$$

де h – кількість символів в рядку, $h = k + r$;

r – кількість перевірочних символів.

При такій організації передачі даних і пакету помилок, довжина якого не перевищує k символів, допускається пошкодження одного або двох сусідніх символів

$$\left(x_1^1, x_1^2, \dots, x_1^h\right), \left(x_2^{1*}, x_2^{2*}, \dots, x_2^h\right), \left(x_k^1, x_k^2, \dots, x_k^h\right), \left(x_{k+1}^1, x_{k+1}^2, \dots, x_{k+1}^h\right),$$

або

$$\left(x_1^1, x_1^2, \dots, x_1^h\right), \left(x_2^1, x_2^2, \dots, x_2^{h*}\right), \left(x_k^{1*}, x_k^{2*}, \dots, x_k^h\right), \left(x_{k+1}^1, x_{k+1}^2, \dots, x_{k+1}^h\right).$$

На приймальній стороні на основі прийнятих пакетів формується початковий масив даних, в якому пошкоджені символи попадуть в різні блоки, за якими здійснюється виявлення та виправлення помилок.

Наприклад, при восьми інформаційних і двох перевірочних символах матриця даних має вигляд:

$$A = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{17} & x_{1j} & y_{11} & y_{12} \\ x_{21} & x_{22} & x_{23} & \dots & x_{27} & x_{2j} & y_{21} & y_{22} \\ x_{31} & x_{32} & x_{33} & \dots & x_{37} & x_{3j} & y_{31} & y_{32} \\ x_{41} & x_{42} & x_{43} & \dots & x_{47} & x_{4j} & y_{41} & y_{42} \\ x_{51} & x_{52} & x_{53} & \dots & x_{57} & x_{5j} & y_{51} & y_{52} \\ x_{61} & x_{62} & x_{63} & \dots & x_{67} & x_{6j} & y_{61} & y_{62} \\ x_{71} & x_{72} & x_{73} & \dots & x_{77} & x_{7j} & y_{71} & y_{72} \\ x_{i1} & x_{i2} & x_{i3} & \dots & x_{i7} & x_{ij} & y_{i1} & y_{i2} \end{pmatrix}.$$

Транспонована матриця зі спотвореними даними має вигляд:

$$B = A^T = \begin{pmatrix} x_{11} & x_{21} & x_{31} & x_{41}^* & x_{51}^* & x_{61}^* & x_{71}^* & x_{81}^* \\ x_{12}^* & x_{22}^* & x_{32}^* & x_{42}^* & x_{52}^* & x_{62}^* & x_{72}^* & x_{82}^* \\ x_{13}^* & x_{23}^* & x_{33}^* & x_{43} & x_{53} & x_{63} & x_{73} & x_{83} \\ x_{14} & x_{24} & x_{34} & x_{44} & x_{54} & x_{64} & x_{74} & x_{84} \\ x_{15} & x_{25} & x_{35} & x_{45} & x_{55} & x_{65} & x_{75} & x_{85} \\ x_{16} & x_{26} & x_{36} & x_{46} & x_{56} & x_{66} & x_{76} & x_{86} \\ x_{17} & x_{27} & x_{37} & x_{47} & x_{57} & x_{67} & x_{77} & x_{87} \\ x_{18} & x_{28} & x_{38} & x_{48} & x_{58} & x_{68} & x_{78} & x_{88} \\ y_{11} & y_{21} & y_{31} & y_{41} & y_{51} & y_{61} & y_{71} & y_{81} \\ y_{12} & y_{22} & y_{32} & y_{42} & y_{52} & y_{62} & y_{72} & y_{82} \end{pmatrix},$$

де x^* – пошкодженні символи.

В приймачі з отриманої матриці B обчислюємо транспоновану матрицю

$$A_1 = B^T = \begin{pmatrix} x_{11} & x_{12}^* & x_{13}^* & x_{14} & x_{15} & x_{16} & x_{17} & x_{18} & y_{11} & y_{12} \\ x_{21} & x_{22}^* & x_{23}^* & x_{24} & x_{25} & x_{26} & x_{27} & x_{28} & y_{21} & y_{22} \\ x_{31} & x_{32}^* & x_{33}^* & x_{34} & x_{35} & x_{36} & x_{37} & x_{38} & y_{31} & y_{32} \\ x_{41}^* & x_{42}^* & x_{43} & x_{44} & x_{45} & x_{46} & x_{47} & x_{48} & y_{41} & y_{42} \\ x_{51}^* & x_{52}^* & x_{53} & x_{54} & x_{55} & x_{56} & x_{57} & x_{58} & y_{51} & y_{52} \\ x_{61}^* & x_{62}^* & x_{63} & x_{64} & x_{65} & x_{66} & x_{67} & x_{68} & y_{61} & y_{62} \\ x_{71}^* & x_{72}^* & x_{73} & x_{74} & x_{75} & x_{76} & x_{77} & x_{78} & y_{71} & y_{72} \\ x_{81}^* & x_{82}^* & x_{83} & x_{84} & x_{85} & x_{86} & x_{87} & x_{88} & y_{81} & y_{82} \end{pmatrix},$$

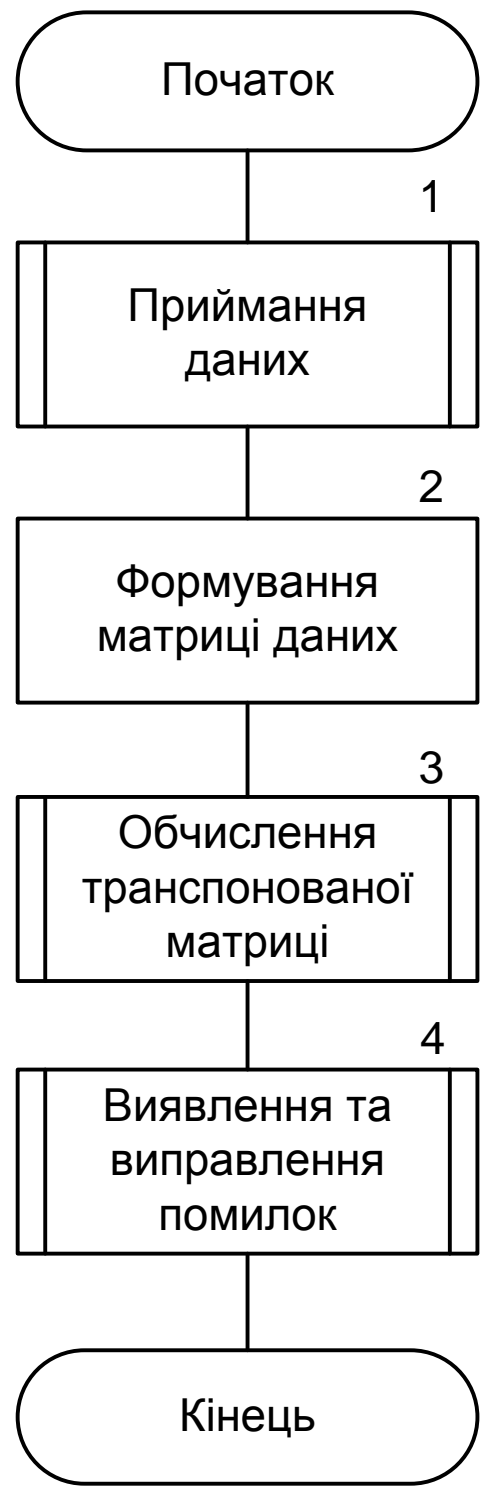
в результаті чого, спотворені символи будуть розміщені в різних рядках і їх можна виправити на основі модулярних коректуючих кодів з використанням лише одного перевірного символу. Таким чином, використавши принцип декомпозиції, можна при одному перевірочному символі виправляти пакети помилок довжиною k , де k – кількість інформаційних символів. При використанні двох перевірочних символів можна виправляти пакети помилок довжиною $2 \cdot k$, так як не більше двох пошкоджених символів будуть знаходитись в одному рядку.

При передачі і прийманні даних на основі декомпозиції (рисунок 3.7) в блоці 1 (рисунок 3.7 а) формується масив даних, які необхідно передати у вигляді матриці розміром $k \times k$.

В блоці 2 по рядках матриці обчислюються перевірочні символи з використанням розроблених модулярних коректуючих кодів [9]. В блоці 3 (рисунок 3.7 а) обчислюється транспонована матриця – замінюються значення рядків матриці на значення стовпців. З рядків транспонованої матриці послідовно формуються пакети, які передаються в канал зв'язку в блоці 5 (рисунок 3.7 а).



а)



б)

Рисунок 3.7 – Алгоритми передачі і приймання даних на основі декомпозиції: а) алгоритм передачі даних; б) алгоритм приймання даних

З прийнятих даних в блоці 1 і в блоці 2 (див. рисунок 3.7 б) формується матриця даних, з якої в блоці 3 обчислюється транспонована матриця, а в

блоці 4 (див. рисунок 3.7 б) виявляються та виправляються помилки по рядках транспонованої матриці.

Таким чином, в підрозділі 3.5 розроблено метод підвищення надійності передачі мультимедійних даних в безпроводних сенсорних мережах, який дозволяє виправляти пакети помилок за рахунок використання декомпозиції пакетів та модулярних коректуючих кодів.

Висновки до третього розділу

1. Розроблено модулярний коректуючий код, який кодує вхідні дані, подані двійковим кодом, та забезпечує гнучкий вибір розрядності і кількості інформаційних символів. Наявність одного перевірного символу забезпечує 100 % виявлення та виправлення помилок в одному інформаційному символі та 87 % виявлення помилок в двох символах.

2. Розроблено метод та алгоритм виправлення багатократних помилок на основі модулярного коректуючого коду, в якому перевірочні символи формуються з використанням наборів різних коефіцієнтів для однакового модуля. Дані коди забезпечують виправлення помилок в t - інформаційних символах при використанні t перевірочних символів, при цьому час формування коду приблизно в 5 разів менший, порівняно з коректуючими кодами СЗК та кодом Ріда-Соломона (127, 87). Апаратні затрати на реалізацію кодування в середньому в 16 разів менші, порівняно з коректуючими кодами СЗК. Використання коректуючого коду в БСМ зменшує відношення E_b/N_0 на 3 дБ при збереженні заданої надійності передачі даних. Розміщення перевірочних символів в кінці поля даних протоколу забезпечує сумісність з мережами стандарту IEEE 802.15.4.

3. Запропонований двовимірний контроль помилок на основі модулярних коректуючих кодів забезпечує виявлення та виправлення k

помилки, які знаходяться в одному рядку або стовпці матриці даних розміру $k \times k$, та $2 \cdot k$ помилок при умові, що вони знаходяться в двох рядках або двох стовпцях матриці даних.

4. Запропоновано метод виявлення та виправлення помилок при виконанні арифметичних операцій (додавання, віднімання, множення) на основі модулярних коректуючих кодів, який дозволяє виявляти помилки в двох десяткових розрядах та виправляти помилки в одному десятковому розряді при використанні одного контрольного символу. Розроблено структурну схему блоку виявлення помилок при виконанні арифметичних операцій.

5. Розроблено метод підвищення надійності передачі даних в безпроводних сенсорних мережах на основі модулярних коректуючих кодів та декомпозиції даних, який забезпечує при одному перевірочному символі виправлення пакетів помилок довжиною k , де k – кількість інформаційних символів. Використання коректуючого коду з двома перевірочними символами забезпечує виправлення пакетів помилок довжиною $2 \cdot k$, оскільки не більше двох пошкоджених символів будуть знаходитись в одному рядку.

РОЗДІЛ 4

КОНЦЕПЦІЯ ПОБУДОВИ ТА АЛГОРИТМИ ПЕРЕДАЧІ ДАНИХ У БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖАХ

4.1 Побудова безпроводних сенсорних мереж на основі колективного інтелекту

Сенсорні мережі з централізованим управлінням мають низьку надійність, пов'язану з можливим виходом з ладу центрального вузла, при цьому мережа припиняє виконувати свої функції. Натомість використання децентралізованого управління мережею забезпечує [68, 204]:

1) розподіл/зменшення вартості. Зменшення вартості здійснюється за рахунок об'єднання та спільного використання ресурсів;

2) об'єднання ресурсів. Кожен вузол в системі виділяє певні ресурси, наприклад, обчислювальну потужність або пам'ять. Об'єднуючи ресурси тисяч вузлів, мережа здатна виконувати складні функції з точки зору складності обчислень. Такими ресурсами може бути пам'ять, щоб зберігати дані, та пропускна спроможність, щоб їх передавати;

3) більша масштабованість/надійність. За відсутності центрального вузла підвищується масштабованість та надійність мережі;

4) висока автономія. Кожен вузол розподіленої системи локально виконує свої функції;

5) динамічність. Середовище розподілених систем є надзвичайно динамічним. Тобто, вузли з'являються та зникають з мережі безперервно. У випадку розподілених обчислень система повинна повторно видавати завдання для обчислення іншим вузлам, щоб гарантувати, що робота не втрачена, якщо з попередніми вузлами втрачено зв'язок.

З дослідженням та формалізацією колективного інтелекту відкрилась теоретична можливість для побудови децентралізованих систем управління [137, 79, 166].

Основна ідея полягає в тому, що для організації взаємодії вузлів безпроводної сенсорної мережі пропонується використати біологічні принципи колективного інтелекту, зокрема мурашиних колоній, що забезпечить ефективне використання ресурсів окремих вузлів та високу живучість безпроводної сенсорної мережі.

Згідно з гіпотезою «розподіленого мозку» в нервовій системі кожної мурашки знаходиться невеликий сегмент центрального мозку, який є колективною власністю колонії і забезпечує існування цієї колонії, як єдиного організму [127]. Крім того, мураха має програму автономної поведінки, яку можна назвати власним сегментом. Оскільки мурашиний мозок складається приблизно з 500 тисяч нейронів, то й обсяг індивідуальної програми, яку може виконувати мураха, буде незначним [127]. Тому, такі програми можуть забезпечувати самостійну поведінку комахи тільки при виконанні елементарних дій і вимагають обов'язкового керуючого сигналу після його закінчення.

За даними дослідників 20% мурашок в колонії не працюють, це так звані «ліниві мурашки» [127]. Однак, є припущення, що «ліниві мурашки» є носіями важливої біологічної інформації для відновлення колонії [131].

Між колонією мурах і безпроводною сенсорною мережею можна провести аналогію: автономний сенсор – окрема мураха; десятки, тисячі автономних сенсорів – колонія мурах.

Гіпотезу «розподіленого мозку» планується використати при розробці методів та алгоритмів розподілу ресурсів автономних сенсорів при виконанні складних задач. При розробці «мурашкових» алгоритмів обміну даними між автономними сенсорами можна досягти ефективного використання обмежених ресурсів останніх для виконання складних задач.

Пропонована концепція побудови БСМ на основі мурашкового інтелекту базується на наступних принципах [91, 212]. Управління БСМ здійснюється розподіленим процесором, при цьому кожний безпроводний вузол мережі є частиною цього розподіленого процесора. Іншими словами, безпроводний вузол виділяє частину своїх обчислювальних ресурсів для функціонування розподіленого процесора (рисунок 4.1). Інша частина ресурсів задіяна на виконання локальних операцій, зокрема збір та обробку даних сенсорів фізичних параметрів. Пам'ять безпроводного вузла розділена на два сегменти: сегмент розподіленого процесора та сегмент локальних команд (рисунок 4.2).

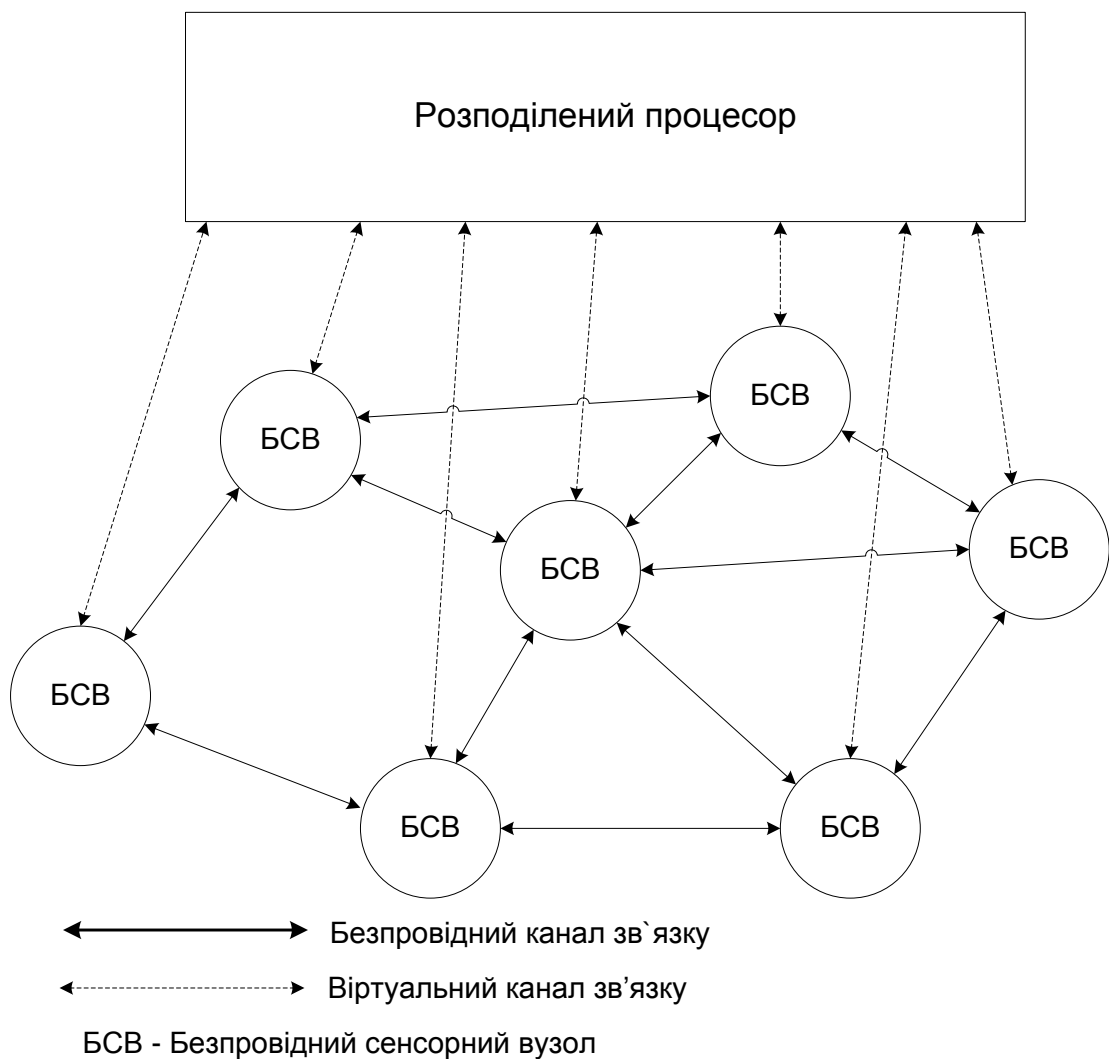


Рисунок 4.1 – Організація розподіленого процесора

Одним із підходів до реалізації загального віртуального адресного простору розподіленого процесора є використання розподіленої пам'яті зі сторінкової організацією [53]. Розподілена пам'ять дозволяє задіяти віртуальну пам'ять кожного безпроводного вузла для підтримки загального віртуального адресного простору. Принцип роботи розподіленої пам'яті наступний. В системі з розподіленою пам'яттю адресний простір поділений на сторінки (4 або 8 Кбайт), які розподілені по безпроводних вузлах мережі. Коли процесор здійснює адресацію до пам'яті, яка не є локальною, відбувається внутрішнє переривання, операційна система зчитує в локальну пам'ять сторінку, яка містить вказану адресу, і перезапускає виконання інструкції, яка викликала переривання і тепер виконується. Це сторінкова організація пам'яті, при якій в якості тимчасового місця зберігання інформації використовується не локальна пам'ять, а віддалена оперативна пам'ять.

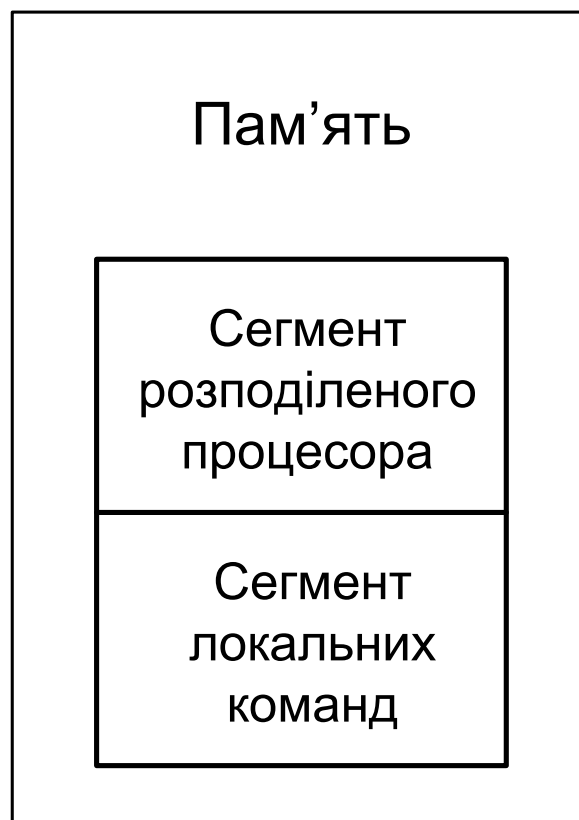


Рисунок 4.2 – Організація пам'яті вузла БСМ

Функціонування розподіленого процесора забезпечується системою елементарних комунікацій між вузлами мережі (рисунок 4.3). Розподілений процесор – спеціалізоване програмне забезпечення, яке фізично знаходиться і виконується у вузлах БСМ.

Розподілений процесор відповідає за виконання наступних функцій: управління топологією, маршрутизацією, енергоресурсами вузлів, розподілом ресурсів, балансуванню навантаженням в мережі та ін.[200, 201].

Крім функцій оптимального управління мережею, розподілений процесор повинен вміти розбивати одну велику задачу (наприклад, обробка зображень) на безліч підзадач, розподіляти ці підзадачі в обчислювальних вузлах, приймати результати обчислень і об'єднувати їх в єдине ціле.

Для підвищення надійності комп'ютерних систем використовують три типи надлишковості – інформаційну, часову та фізичну [204]. При інформаційній надлишковості до повідомлення додаються додаткові символи, які дозволяють виявляти та виправляти помилки (коректуючи коди) [152]. При часовій надлишковості вже виконану дію при необхідності повторюють.

При фізичній надлишковості в систему додається додаткове обладнання або процеси, які забезпечують роботу системи при втраті або непрацездатності окремих вузлів мережі. В БСМ фізична надлишковість, може бути як апаратною, так і програмною. Так, наприклад, для забезпечення програмної надлишковості до системи необхідно додати процеси, які б відновлювали роботу мережі при збої окремих процесів.

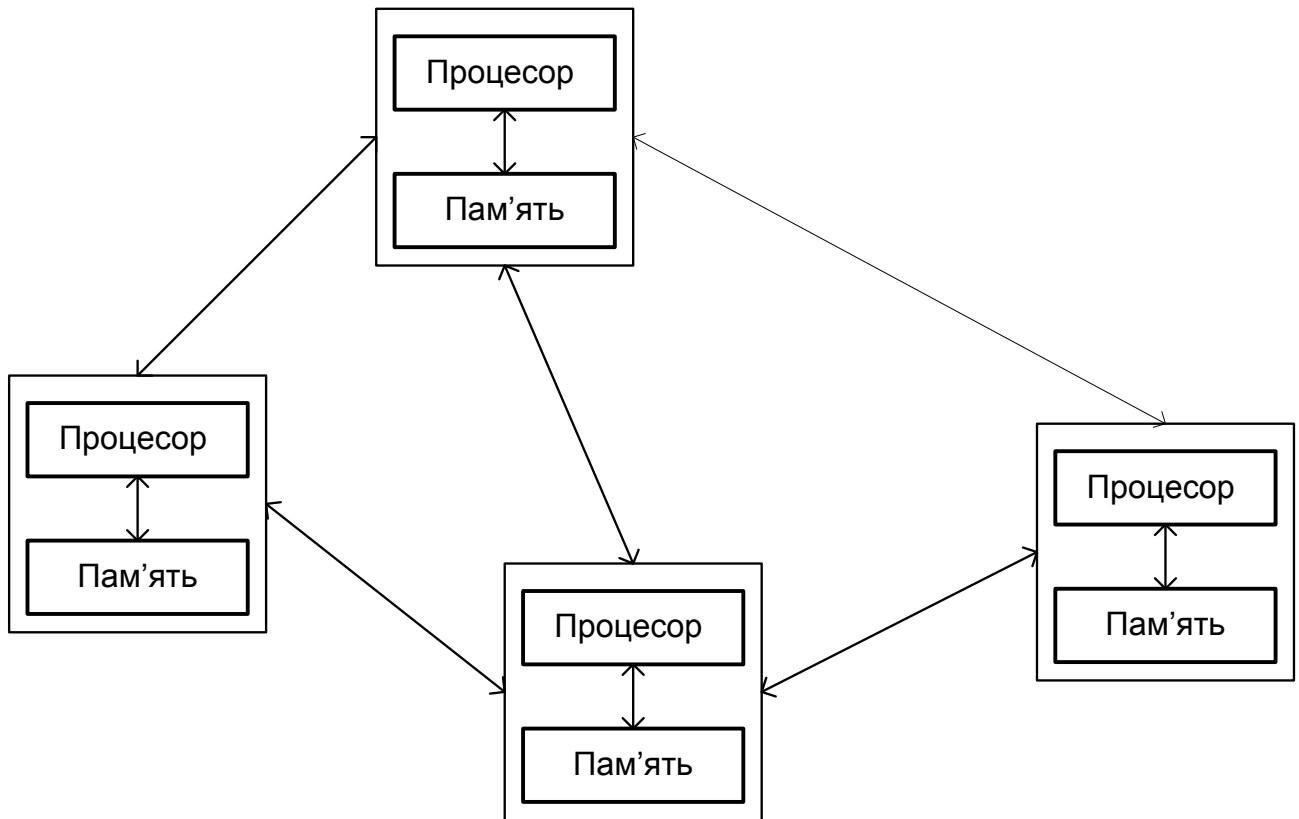


Рисунок 4.3 – Комунікації між вузлами БСМ

Для реалізації фізичної надлишковості за принципом мурашиних колоній в пропонованій концепції БСМ частина вузлів приблизно 20%, в яких розміщені особливо важливі програми і дані розподіленого процесора, будуть працювати в режимі економії енергії, що дасть можливість відновити роботу мережі після виходу з ладу частини вузлів. Відновлення необхідних сегментів розподіленого процесора відбувається шляхом реконфігурації безпроводних вузлів.

4.1.1 Оцінка живучості безпроводних сенсорних мереж

Живучість – здатність системи виконувати свої основні функції (з допустимою втратою якості функціонування) при дії негативних факторів зовнішнього середовища або несприятливих умов експлуатації [59, 14].

До втрати живучості БСМ призводить відмова вузлів, яка може бути пов'язана з поломкою компонентів вузла або розрядкою батареї живлення та відсутністю зв'язку між вузлами. Враховуючи високу надійність електронних компонентів, втрата живучості БСМ відбувається при вилученні зі структури мережі вузлів, які здійснюють ретрансляцію пакетів і розміщені на перетині найкоротших маршрутів. Живучість мережі вважають високою, якщо необхідно вилучити значну кількість вузлів і/або ребер, щоб істотно погіршити її функціонування.

Важливість вузла в мережі визначається [14]

$$b_m = \sum_{i \neq j} \frac{B(i, m, j)}{B(i, j)},$$

де $B(i, j)$ – загальна кількість найкоротших маршрутів між вузлами i і j ;

$B(i, m, j)$ – кількість найкоротших маршрутів між вузлами i і j , які проходять через вузол m .

Для оцінки живучості БСМ використаємо функціонал, який визначає середню кількість робочих маршрутів [14]:

$$\varphi_1(P(t/S)) = \sum_{i=1}^{N_0} P_i(t/S)(N_0 - 1),$$

де $P_i(t/S)$ – ймовірність того, що в момент t неробочими є i маршрутів;

N_0 – загальна кількість маршрутів в мережі;

та ймовірність того, що кількість робочих маршрутів не менша заданої [14]:

$$\varphi_2(P(t/S)) = \sum_{i=1}^{i_0} P_i(t/S),$$

де i – максимально допустима кількість неробочих маршрутів.

Отже, для забезпечення заданої живучості БСМ необхідно, щоб середня кількість робочих маршрутів була більша за мінімально необхідну.

Побудова безпроводних сенсорних мереж на основі розробленої концепції дозволить підвищити ефективність використання обмежених апаратних ресурсів безпроводних вузлів при виконанні складних завдань. Висока живучість мережі досягається за рахунок резервування важливих сегментів програмного забезпечення розподіленого процесора в безпроводних вузлах.

4.2 Протокол передачі даних на основі мережного кодування

На основі запропонованого в розділі 2 методу мережного кодування даних в системі залишкових класів розроблено структуру протоколу передачі даних.

В процесі реалізації алгоритму (рисунок 4.4) в блоці 1 відбувається вибір взаємно простих модулів та їх розподіл для всіх вузлів мережі [95, 84]. На наступному кроці з отриманого повідомлення обчислюються залишки за вибраними модулями (блок 2), які передаються знайденими маршрутами (блок 3).

За умови, що наступний вузол (блок 4) мережі є незалежним (вузол, який ретранслює дані тільки від одного джерела), по даному маршруту передаються залишки більшої розрядності, в іншому випадку дані обробляються на проміжному вузлі (блок 6) і передаються (блок 7) наступним вузлам в мережі.

Таким чином, дані (залишки) поступають на кінцевий вузол (блок 8) (базову станцію) і відновлюються в блоці 9 в позиційну систему числення.

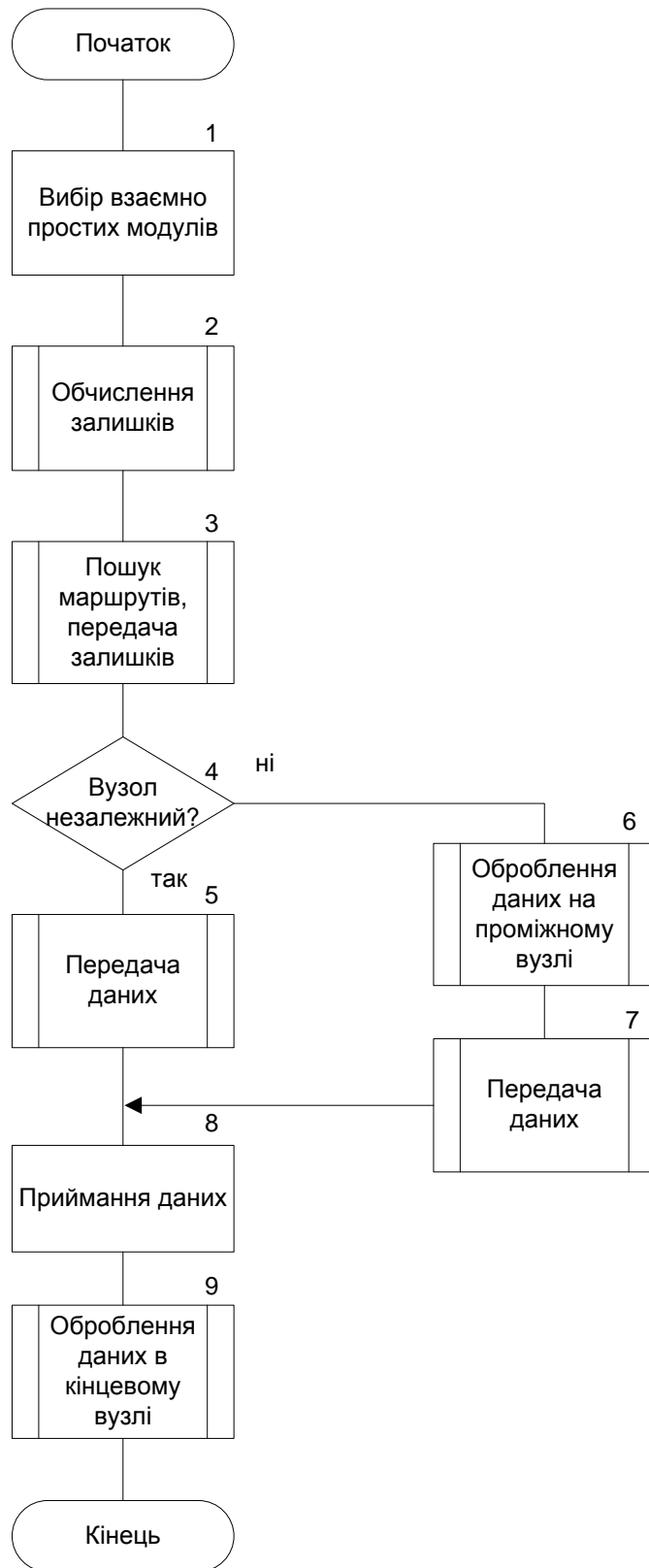


Рисунок 4.4 – Алгоритм мережного кодування в системі залишкових класів

Для розробки структури протоколу необхідно розглянути обмін даними між координатором і кінцевим пристроєм. Обмін повідомленнями між

кінцевим пристроєм і координатором в БСМ стандарту ZigBee виконується без використання кадрів – маяків на основі послідовності операцій (рисунок 4.5). Використовуючи схему CSMA-CA, кінцевий пристрій посилає інформаційний кадр координатору. Координатор відправляє кадр підтвердження (опційно) про успішну доставку даних.



Рисунок 4.5 – Обмін даними з координатором без використання маяків

Загальний формат кадру управління доступом до середовища (MAC кадру) складається з поля MAC-заголовок (MHR), поля даних MAC і завершальної частини MAC-кадру (MFR) (рисунок 4.6).

Поле управління кадром займає 2 байти і містить інформацію, що визначає тип кадру, адресні поля та інші поля управління (рисунок 4.7).

В протоколі ZigBee використовуються чотири типи кадру, яким відповідають коди (000 – 011) (таблиця 4.1).

Байтів:2	1	0/2	0/2/8	0/2	0/2/8	Змінне	2
Упра - вління кадром	Поряд – ковий номер	PAN ID - одержувача	Адреса одержувача	PAN ID - відправника	Адрес відправника	Поле даних	FCS
Адресні поля						Поле даних MAC	MFR
MHR							

Рисунок 4.6 – Загальний формат MAC кадру: FCS – контрольна сума кадру

В розробленому протоколі модифіковано субполе «тип кадру», яке має 3 біти, і за рахунок зарезервованих в стандартному протоколі кодів (100-111) вказується, що при формуванні пакетів використовується мережне кодування даних (таблиця 4.1).

Поле даних містить три типи субполів, зокрема, субполе номеру кадру даних SN1, SN2, значення модуля, за яким обчислено залишки та значення залишків – b_{ij} . (рисунок 4.8). При цьому, якщо пакет даних містить залишки за одним модулем, то формується пакет даних (рисунок 4.9 а), а при об'єднанні пакетів даних формується пакет даних (рисунок 4.9 б).

Біт: 0-2	3	4	5	6	7-9	10-11	12-13	14- 15
Тип кадру	Безпека включена	Затримка кадру	Запит підтвердження	Компресія PAN ID	Резерв	Режим: адреса одержувача	Версія кадру	Режим: адреса відправника

Рисунок 4.7 – Поле управління кадру

Таблиця 4.1 – Значення субполя тип кадру

Тип кадру	Опис
000	Маяк
001	Дані
010	Підтвердження
011	MAC команда
100	Мережне кодування (залишки за одним модулем)
101	Мережне кодування (залишки за двома модулями)
110-111	Зарезервовані

Дослідження розробленого протоколу мережного кодування даних проведено в середовищі Riverbed Modeler Academic Editor 17.5 –

віртуального середовища для моделювання та оцінки ефективності комп'ютерних мереж з фіксованими та мобільними вузлами, в тому числі маршрутизаторів, комутаторів, протоколів, серверів і окремих додатків [150].

Riverbed Modeler містить бібліотеки вже реалізованих протоколів безпроводних комп'ютерних мереж, зокрема ZigBee [183].



а)



б)

Рисунок 4.8 – Формат поля даних кадру: а) передача залишків за одним модулем: б) передача залишків за двома модулями: SN1, SN2 – номер кадру даних; M – значення модуля; bij – залишки; MM – добуток модулів

Параметри моделювання ZigBee мережі наведено в таблиці 4.2.

В процесі дослідження протоколу використано чотири моделі процесу в середовищі OPNET ZigBee:

- 1) ZigBee MAC- модель, яка реалізує модель протоколу IEEE 802.15.4 MAC, зокрема, сканування каналів, відновлення процесу роботи протоколу.
- 2) ZigBee модель програми, яка генерує і приймає трафік, а також формує звіти моделювання.
- 3) ZigBee модель доступу до середовища (CSMA/CA), яка реалізує протокол доступу до середовища MAC рівня.
- 4) ZigBee модель мережі, яка реалізує мережевий рівень ZigBee і відповідає за маршрутизацію трафіку та процес приєднання до мережі [64].

В таблиці 4.3 наведено параметри налаштування вузлів мережі на фізичному та мережевому рівнях. Конфігурація координатора наведена на рисунку 4.9.

Таблиця 4.2 – Параметри моделювання ZigBee мережі

Назва	Значення
Кількість вузлів	30
Кількість координаторів	1
Кількість маршрутизаторів	4
Кількість кінцевих вузлів	25
Час моделювання	10800 с

Таблиця 4.3 – Конфігурація вузлів мережі

Назва	Значення
Фізичний рівень	
Частота передачі	2,4 ГГц
Потужність передавача	0,005 Вт
Чутливість приймача	-85 дБм
Швидкість передачі	Автоматичне обчислення
MAC рівень	
Підтвердження отримання пакету	Включено
Час очікування підтвердження	0,05 с
Кількість повторних передач пакету	5
Мінімальне значення очікування передачі в алгоритмі CSMA-CA	3
Максимальна кількість відмов в алгоритмі CSMA-CA	5
Мережевий рівень	
Функції вузла	Координатор

Продовження табл.4.2

Розподілу інтервалу часу між пакетами	Псевдовипадковий (нормальний розподіл)
Розмір пакету	800 біт
Час початку передачі	Псевдовипадковий
Час зупинки	Нескінченість

Досліджувана мережа (див. таблицю 4.2) складається із 30 вузлів, з яких один координатор, чотири маршрутизатори та 24 кінцеві пристрої (рисунок 4.10). Розроблено два сценарії роботи мережі: scenario1 використовує стандартний протокол ZigBee, scenario 2_NC використовує розроблений протокол передачі даних на основі методу мережного кодування.

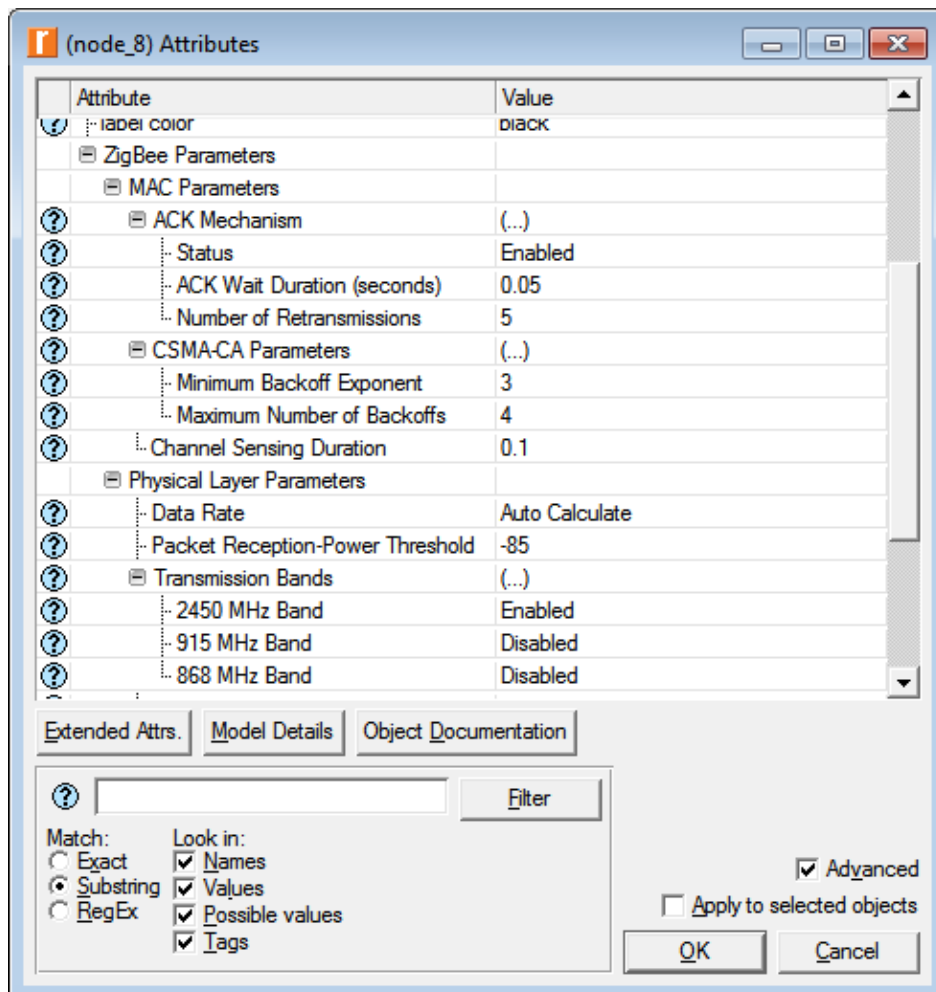


Рисунок 4.9 – Конфігурація координатора

Результати експериментального дослідження пропускної здатності мережі в координаторі (node_8) та маршрутизаторах (node_2, node_3, node_4, node_5) наведені відповідно на рисунках 4.11 – 4.15. Їх аналіз показав наступне:

- при використанні протоколу мережного кодування загальна пропускна здатність мережі в середньому збільшилася на 54% (рисунок 4.15);
- на окремих маршрутизаторах (node_2 – node_5) пропускна здатність мережі зростає на 25 % – 60 % (рисунок 4.11 – 4.15).

Експериментально оцінено затримку передачі пакетів (рисунок 4.116), яка збільшується приблизно на 9%.

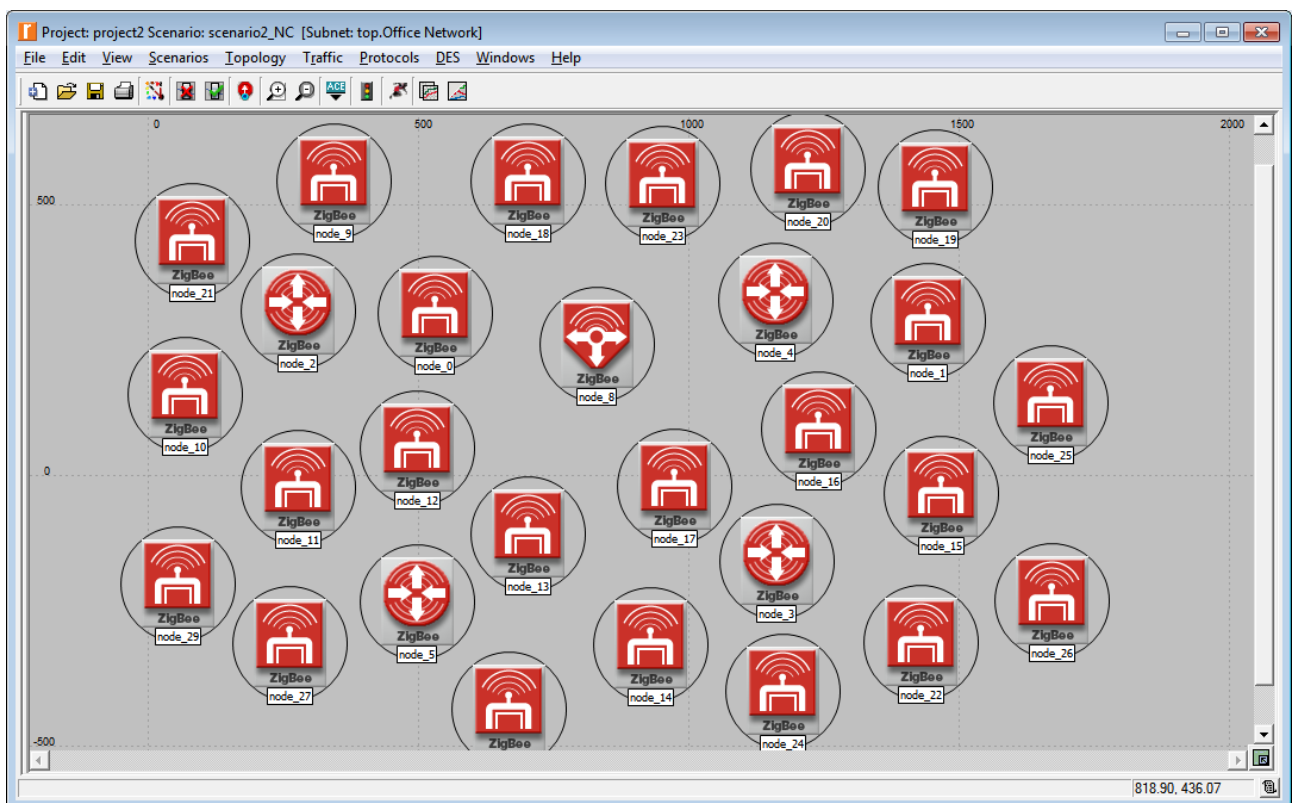


Рисунок 4.10 – Топологія досліджуваної мережі

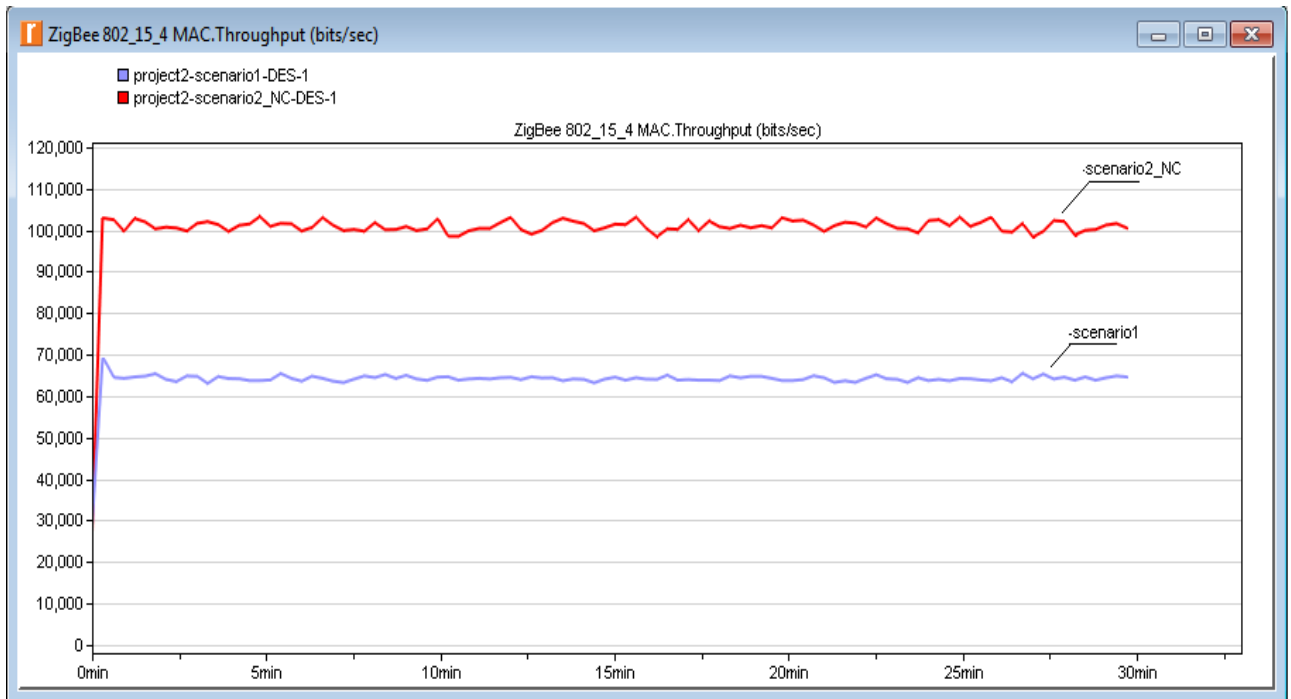


Рисунок 4. 11 – Пропускна здатність мережі (координатор)

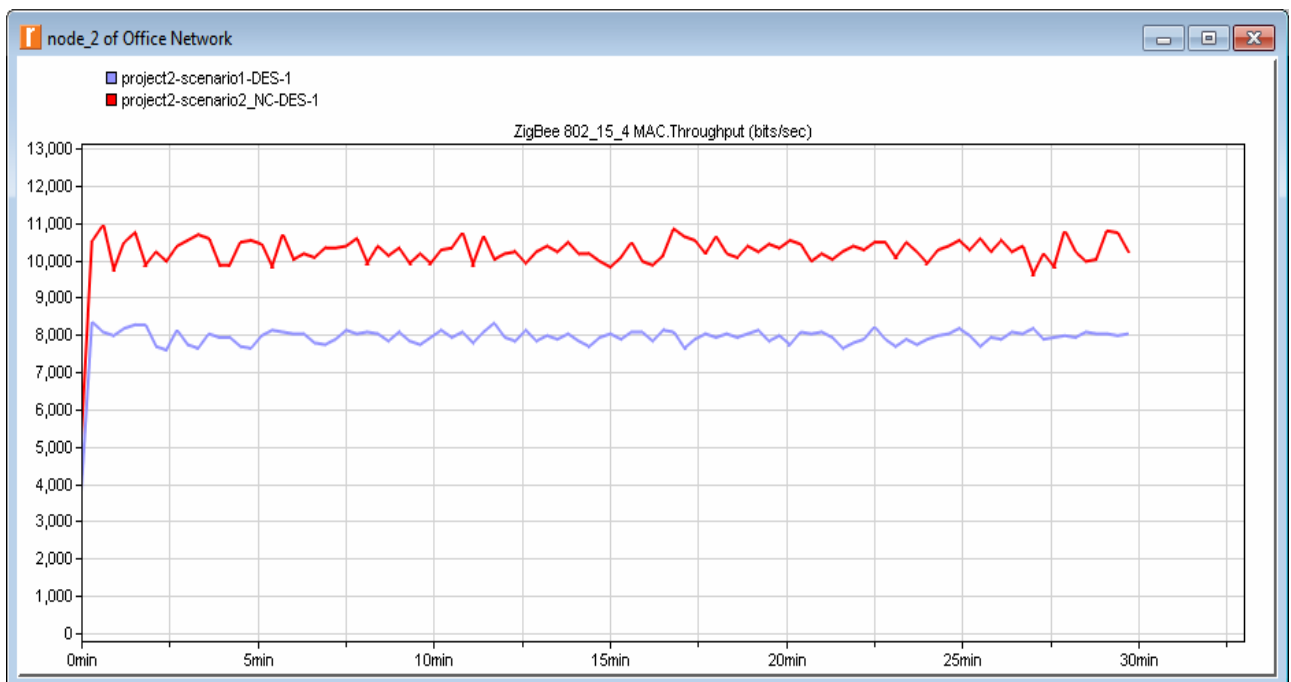


Рисунок 4.12 – Пропускна здатність мережі у вузлі 2

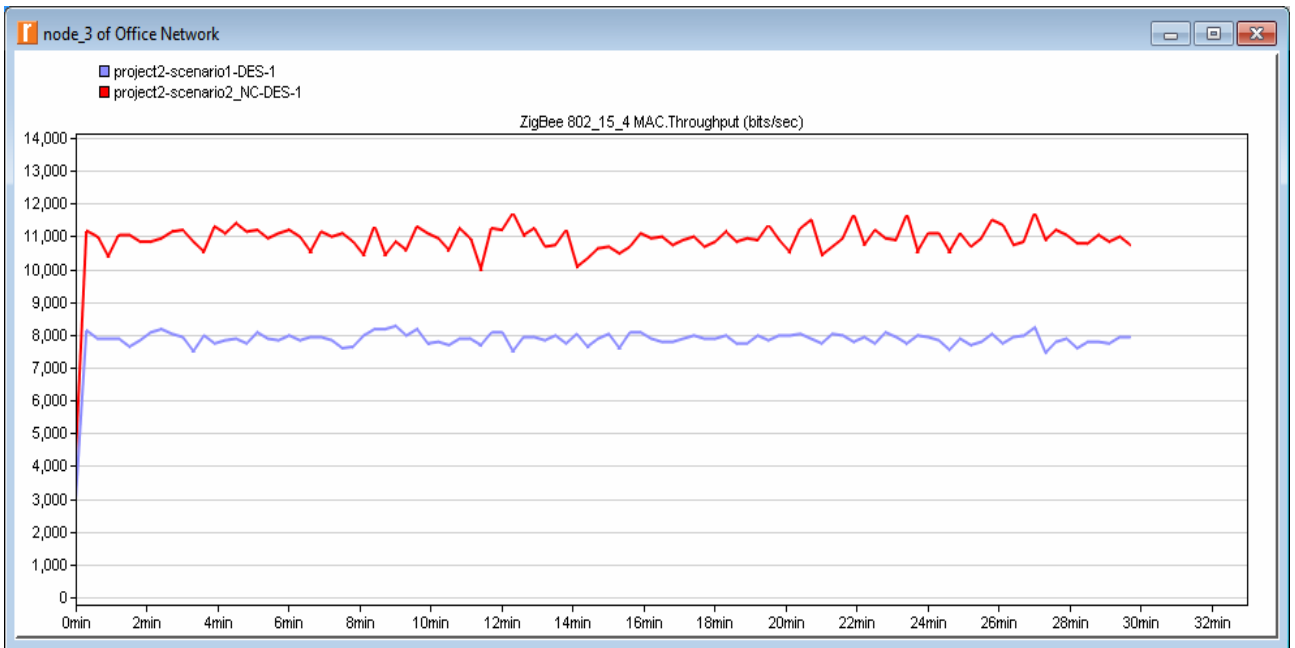


Рисунок 4.13 – Пропускна здатність мережі у вузлі 3

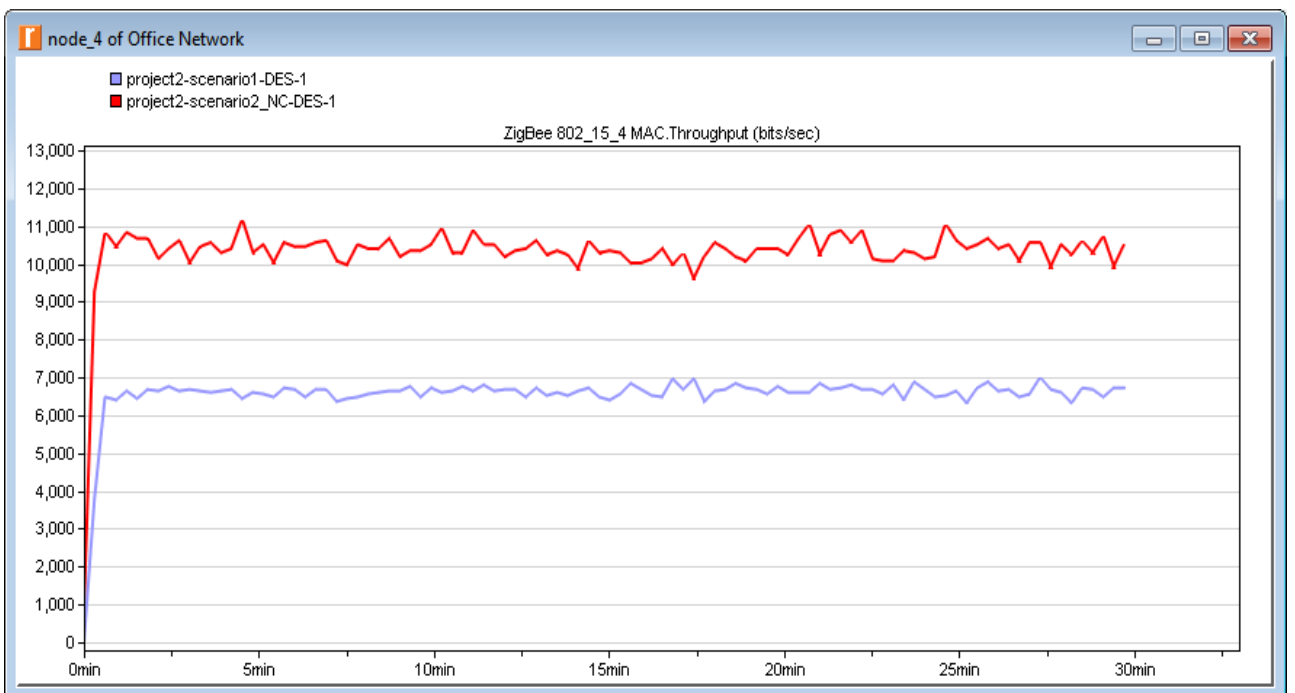


Рисунок 4. 14 – Пропускна здатність мережі у вузлі 4

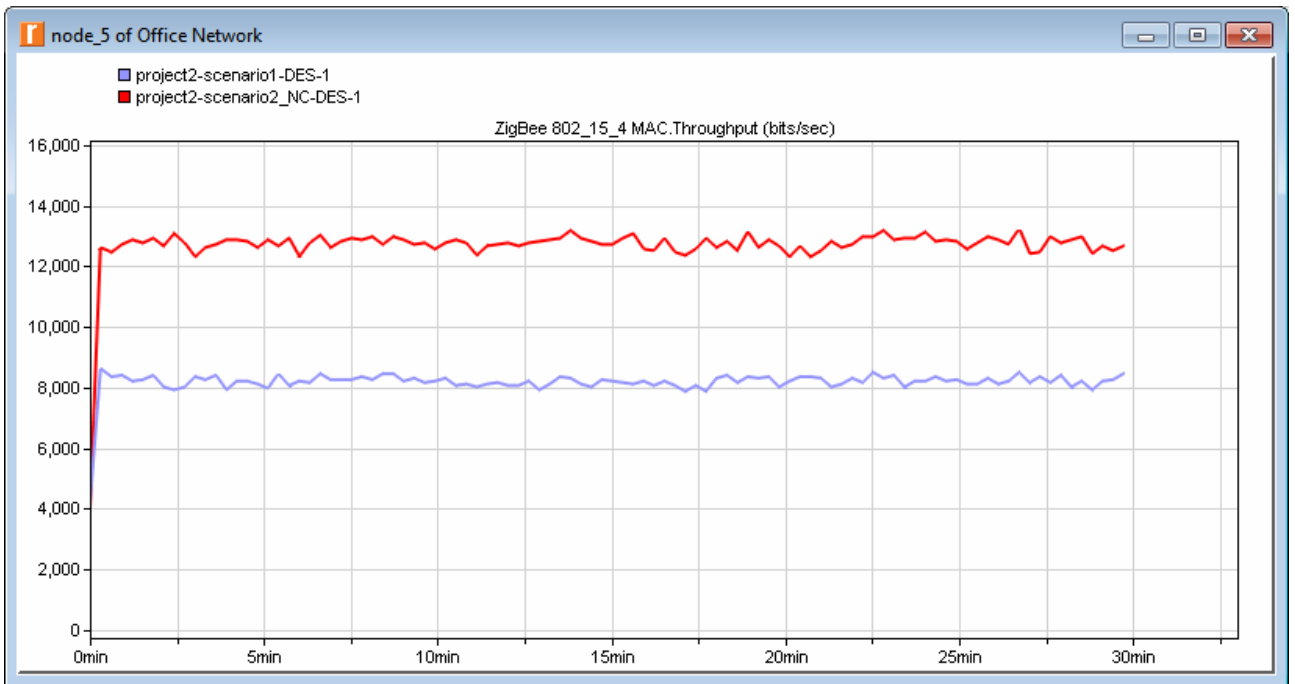


Рисунок 4.15 – Пропускна здатність мережі у вузлі 5

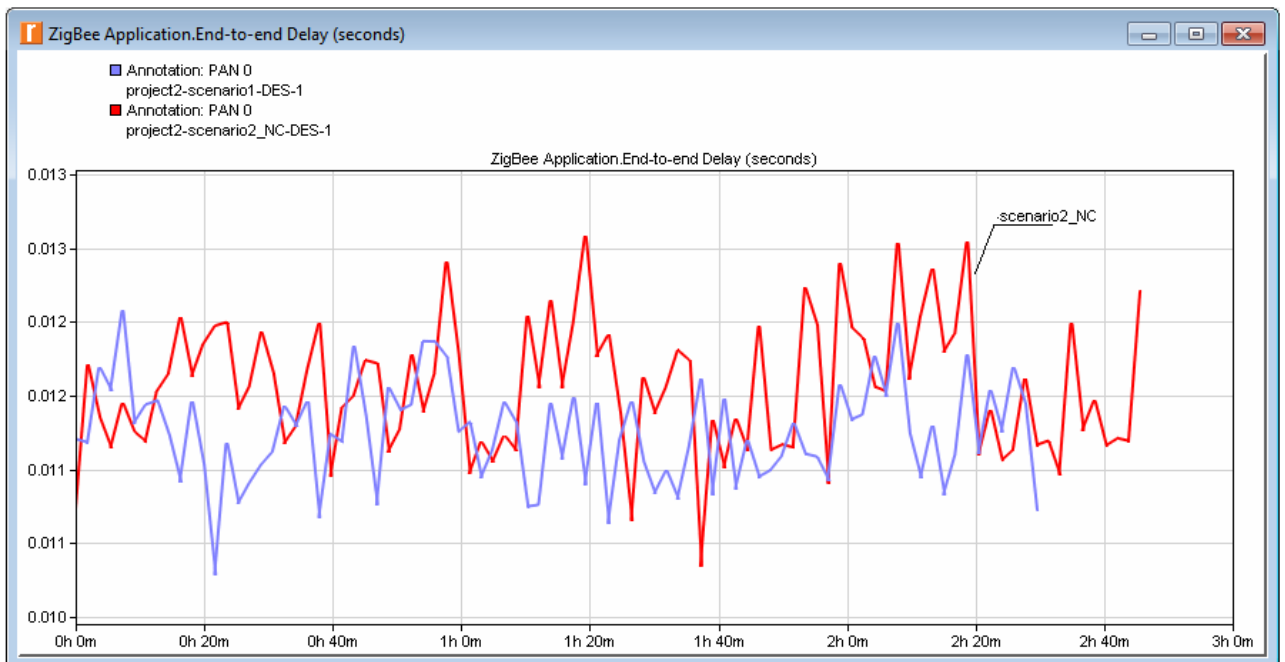


Рисунок 4.16 – Затримка передачі пакетів в мережі

Проведені дослідження пропускної здатності безпроводної сенсорної мережі показали, що використання розробленого протоколу мережного кодування підвищує пропускну здатність мережі в середньому на 54% при незначному (9 %, або 0,001 с) збільшенні затримки передачі пакетів.

Таким чином, в даному підрозділі розроблено та експериментально досліджено протокол передачі даних на основі мережного кодування в системі залишкових класів, який дозволяє підвищити пропускну здатність мережі в середньому на 54% при незначному (9 %, або 0,001 с) збільшенні затримки передачі пакетів.

4.3. Алгоритм адаптивної передачі даних у безпроводних сенсорних мережах

Недоліком відомих адаптивних алгоритмів передачі даних, які враховують стан маршрутів, є відсутність можливості зміни коректуючих властивостей кодів в процесі роботи системи передачі, що призводить до збільшення надлишковості при передачі даних і, відповідно, до зменшення корисної пропускну здатності за рахунок повторної передачі пакетів або передачі надлишкових даних для виявлення та виправлення помилок.

Для усунення вище наведених недоліків розроблено алгоритм передачі даних на основі модулярних коректуючих кодів (рисунок 4.17).

Вибір параметрів коректуючих кодів здійснюється в залежності від значення індикатора якості зв'язку (LQI) і відповідає таблиці 4.4. Після чого сформований пакет (дані + перевірочні символи) поступають на блок передачі даних блок 4 (рисунок 4.17) [6].

Під час наступних сеансів передачі даних при виявленні декількох альтернативних маршрутів з потрібним вузлом мережі маршрутизатор використовує історію з'єднання (аналізує значення LQI попередніх з'єднань) (блок 2) і вибирає маршрут проходження пакету з найменшою ймовірністю пошкодження даних і необхідними параметрами модулярного коректуючого коду (блок 3). Вимірювання LQI здійснюється для кожного прийнятого пакету, і результат у вигляді цілого числа в діапазоні від 0x00 до 0xff

передається до MAC-підрівня. Мінімальні і максимальні значення LQI (0x00 і 0xff) відповідають найнижчій і найвищій якості з'єднання.

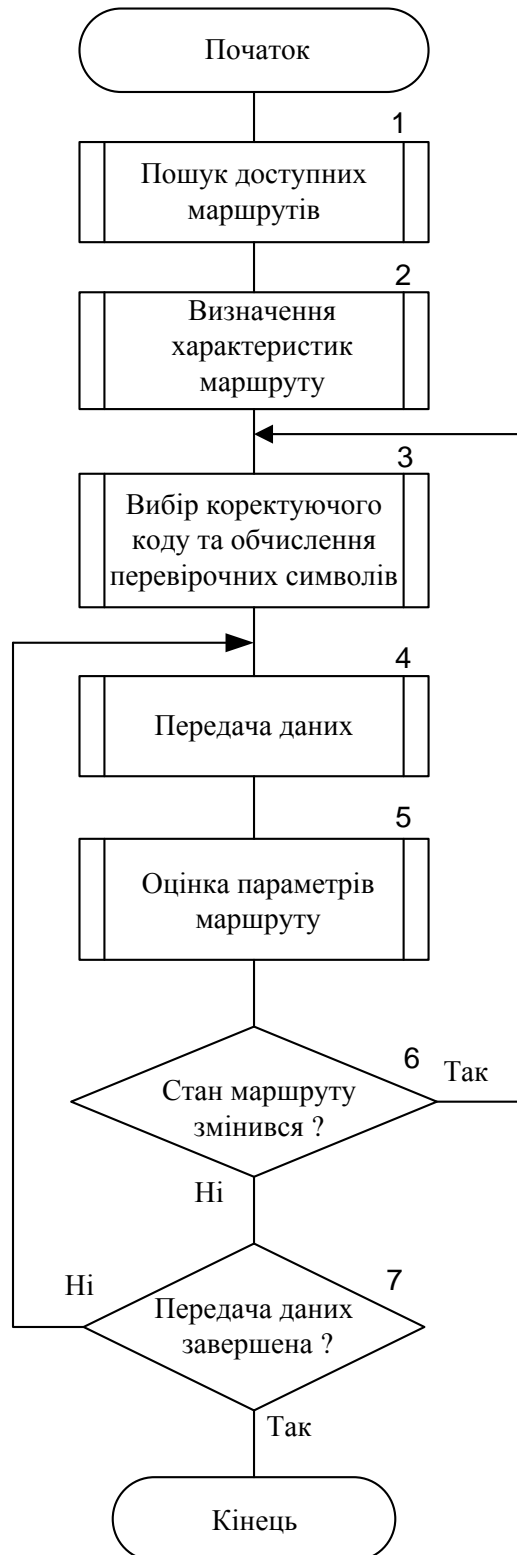


Рисунок 4.17 – Алгоритм адаптивної передачі даних

Діапазон значень LQI рівномірно розподіляється між верхньою і нижньою межами (0x00 і 0xff). При цьому використовуються щонайменше вісім унікальних значень LQI.

Таблиця 4.4 – Параметри коректуючого коду при різних значеннях LQI

№	Значеннях LQI	Швидкість коду R	Кількість символів, що можна виправити
1	225 - 255	Виявлення помилок	
2	193-224	0,79	1
3	161-192	0,65	2
4	129-160	0,56	3
5	97-128	0,48	4
6	65-96	0,43	5
7	33-64	0,39	6
8	0-32	0,35	7

В процесі передачі даних передавач постійно оцінює значення LQI маршруту (блок 5), і при його зміні (блок 6), відповідно до таблиці 4.4, відбувається адаптивна зміна параметрів коректуючого коду. Після закінчення передачі масиву даних (блок 7) алгоритм відновлює свою роботу з пошуку доступних маршрутів (блок 1).

Висновки до четвертого розділу

1. Розроблено концепцію побудови безпроводних сенсорних мереж, яка базується на принципах колективного інтелекту та розподіленого процесора, забезпечує виділення частини обчислювальних ресурсів безпроводних сенсорів для функціонування розподіленого процесора та високу живучість

мережі за рахунок резервування важливих сегментів програмного забезпечення розподіленого процесора в безпроводних вузлах та децентралізованого управління мережею.

2. Розроблений протокол передачі даних на основі мережного кодування даних забезпечує підвищення пропускної здатності мережі, порівняно з протоколом стандарту IEEE 802.15.4, в середньому на 54% при незначному (9 %, або 0,001 с) збільшенні затримки передачі пакетів.

3. Розроблено алгоритм кодування та передачі даних в безпроводних сенсорних мережах на основі модулярних коректуючих кодів, який здійснює адаптивну зміну коректуючих властивостей коду в залежності від стану маршруту передачі даних.

РОЗДІЛ 5.

СТРУКТУРНА ОРГАНІЗАЦІЯ ТА СИНТЕЗ КОМПОНЕНТІВ СЕНСОРНИХ МЕРЕЖ НА ПРОГРАМОВАНИХ ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМАХ

Підвищення швидкодії апаратних засобів обробки даних можна досягти завдяки ефективним методам і алгоритмам обробки та новій технології виготовлення апаратних засобів, зокрема, використанню апаратних засобів з вищою тактовою частотою. Проблема підвищення швидкодії особливо актуальна при обробці мультимедійних даних, які характеризуються великими обсягами та чутливістю до затримки, а також при обробці даних в протоколах передачі телекомунікаційних мереж, зокрема, при застосуванні коректуючих кодів. Оскільки обробка даних часто пов'язана з виконанням великої кількості арифметичних операцій, то від ефективності реалізації останніх в значній мірі залежить ефективність обробки в цілому. Як відомо, виконання арифметичних операцій в системі залишкових класів (СЗК) має ряд переваг, які досягаються за рахунок можливості паралельного виконання арифметичних операцій, малої (3-6 біт) розрядності залишків та незалежності залишків [2, 182].

Основною елементною базою для реалізації систем обробки даних на даний час є мікропроцесори, мікроконтролери, програмовані логічні інтегральні схеми (ПЛІС) та системи на кристалі (System-on-a-Chip, SoC). Ефективність виконання арифметичних операцій в СЗК на ПЛІС в значній мірі залежить від методів та алгоритмів їх виконання. В дисертаційній роботі у якості критеріїв оцінки ефективності виконання арифметичних операцій вибрано апаратні затрати (кількість логічних елементів) та час виконання відповідних операцій (максимальна затримка проходження сигналу).

Для ефективної реалізації компонентів БСМ на основі запропонованих в попередніх розділах методів необхідно дослідити методи виконання

арифметичних операцій в модулярній арифметиці при їх реалізації на ПЛІС [82, 104].

5.1 Експериментальні дослідження апаратної та часової складності методів виконання модулярних операцій

5.1.1 Операції додавання та віднімання в системі залишкових класів

Розглянемо виконання основних арифметичних операцій в СЗК [2, 82 182]

– додавання в СЗК

$$\begin{cases} c_i = a_i + b_i, & \text{при } a_i + b_i < p_i \\ c_i = a_i + b_i - p_i, & \text{при } a_i + b_i \geq p_i \end{cases}; \quad (5.1)$$

– віднімання в СЗК

$$\begin{cases} c_i = a_i - b_i, & \text{при } a_i - b_i \geq 0 \\ c_i = a_i - b_i + p_i, & \text{при } a_i - b_i < 0 \end{cases}; \quad (5.2)$$

– множення в СЗК

$$\begin{cases} c_i = a_i \times b_i, & \text{при } a_i \times b_i < p_i \\ c_i = (a_i \times b_i) \bmod p_i, & \text{при } a_i \times b_i \geq p_i \end{cases}. \quad (5.3)$$

Використовуючи формули (5.1, 5.2) та оператори мови Verilog, арифметичні операції додавання та віднімання в СЗК можна записати різними способами, наприклад [82]:

спосіб М.1: додавання – `assign c=(a+b)%p;`

віднімання – `assign c=(a-b)%p;`

спосіб M.2: додавання – $\text{assign } c = ((a+b) < m) ? (a+b) : ((a+b) - p);$

віднімання – $\text{assign } c = ((a-b) >= 0) ? (a-b) : ((a-b) + p);$

де a, b – доданки; p – модуль.

Проведені дослідження розглянутих способів опису операції модулярного додавання мовою Verilog показали, що спосіб M2 є більш ефективним з точки зору апаратних затрат і часу виконання операції (рисунок 5.1, рисунок 5.2) відповідно.

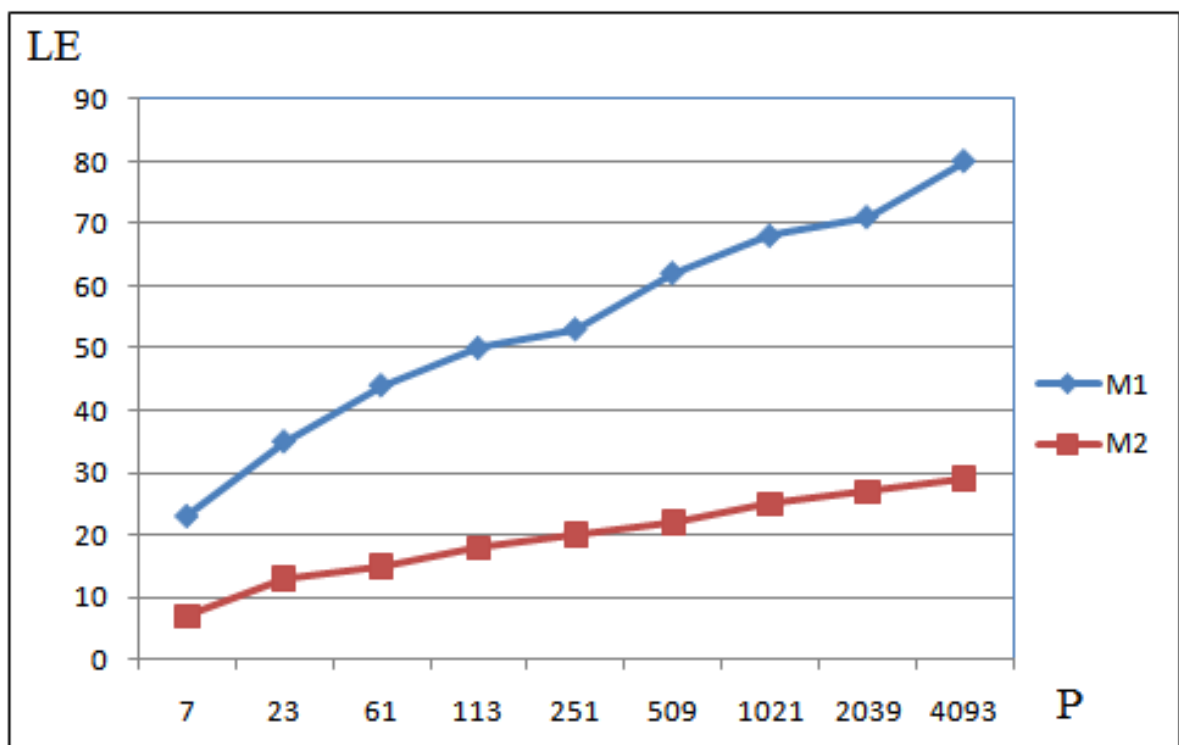


Рисунок 5.1 – Залежність апаратних затрат від значення модуля при реалізації модулярного додавання способами M1 і M2

Як видно з рисунку 5.1, апаратні затрати на виконання модулярного додавання способом M2 при різній розрядності модулів в 2,5-3,2 рази менші порівняно із застосуванням спеціальних засобів мови Verilog. При цьому швидкодія способу M2 в 1,2-1,7 разів вища за спосіб M1 (рисунок 5.2).

В таблицях 5.1, 5.2 наведені результати досліджень апаратної складності та швидкодії виконання операцій додавання та віднімання в СЗК

для різних наборів модулів та діапазонів подання чисел, де спосіб 1 – опис на мові Verilog з використанням оператора отримання залишку %, спосіб 2 – опис на мові Verilog операцій додавання і віднімання на основі формул 5.1 і 5.2. При дослідженні використовувалась різна кількість модулів та їхні значення.

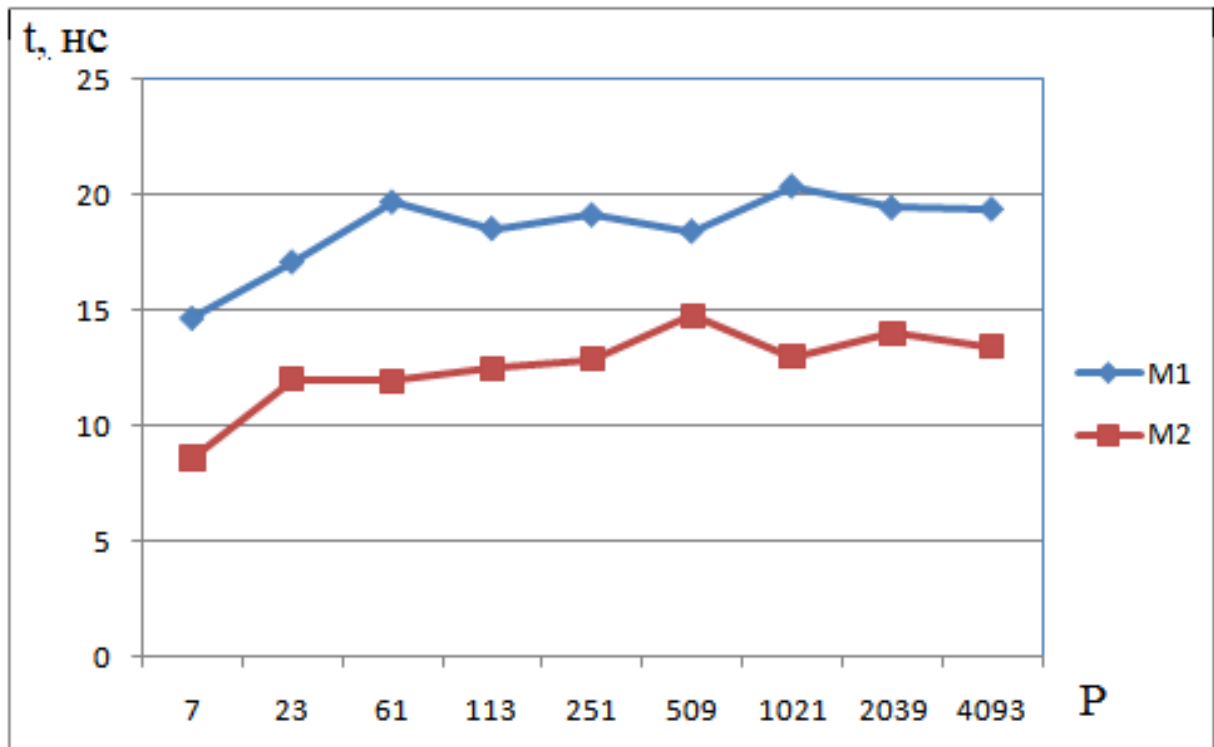


Рисунок 5.2– Залежність часу затримки від значення модуля при реалізації модулярного додавання методами M1 і M2

Як видно з таблиць 5.1, 5.2, застосування способу 2 для опису на мові Verilog на основі формул (5.1, 5.2) забезпечує зменшення кількості логічних елементів приблизно в 2,5 - 3 рази та підвищення швидкодії майже в 2 рази.

Таблиця 5.1 – Параметри виконання операції додавання в СЗК

Діапазон подання чисел: 2^n	Значення модулів	Кількість логічних елементів		Час затримки, нс	
		Спосіб	Спосіб	Спосіб	Спосіб
		1	2	1	2
n=8	3, 7, 13	64	24	16,437	8,889
n=16	239, 277	115	42	20,072	13,340
	29, 43, 47	120	43	18,610	11,864
	3, 7, 11, 13, 23	128	48	17,768	11,517
n=24	4091, 4111	160	61	19,779	14,164
	229, 239, 307	168	62	19,156	12,908
	7, 23, 29, 59, 61	181	64	17,371	12,379

Таблиця 5.2 – Параметри виконання операції віднімання в СЗК

Діапазон подання чисел 2^n	Значення модулів	Кількість логічних елементів		Час затримки, нс	
		Спосіб	Спосіб	Спосіб	Спосіб
		1	2	1	2
8	3, 7, 13	48	23	14,916	8,614
16	239, 277	83	36	20,436	10,71
	29, 43, 47	78	37	17,038	9,912
	3, 7, 11, 13, 23	131	44	18,549	11,416
24	4091, 4111	118	52	20,612	11,931
	229, 239, 307	138	53	18,704	12,193
	7, 23, 29, 59, 61	164	56	23,256	11,716

5.1.2 Модулярне множення

Модулярне множення широко використовується в розроблених модулярних коректуючих кодах та методах цифрової обробки даних [102, 106], і тому від ефективності його реалізації залежить ефективність роботи алгоритмів в цілому [82]. Враховуючи, що модулярне множення, порівняно з додаванням і відніманням, є складнішою операцією, розглянемо його реалізацію на ПЛІС більш детально, зокрема наступними п'ятьма методами:

1. Виконується множення двох чисел, а потім за допомогою операції ділення знаходимо залишок у вигляді

$$r_i = (a_i \times b_i) \bmod p_i = (a_i \times b_i) - q \cdot p_i,$$

$$\text{де } q_i = \left\lfloor \frac{a_i \times b_i}{p_i} \right\rfloor.$$

Даний метод є достатньо простим для реалізації засобами мови Verilog, але має значну затримку виконання.

2. Залишок знаходиться за допомогою операції послідовного віднімання модуля: поки $(a_i \times b_i) \geq p_i$, зменшувати $(a_i \times b_i)$ на p_i .

Даний метод доцільно використовувати при розрядності модулів $n < 5$.

3. Операцію множення за модулем p_i можна реалізувати з використанням таблиці множення розміром p_i на p_i , що потребує збереження масиву даних розміром $p \times p$. Зменшення розміру масиву можна досягти використовуючи властивість симетрії вказаних таблиць, при цьому зростають апаратні затрати на реалізацію логічних переходів (розгортання скорочених таблиць). Отже, виконання модулярного множення на основі таблиць є доцільним при розрядності модулів $n < 5$.

4. Метод індексного множення. Метод індексного множення ґрунтується на твердженні: якщо модуль p просте число, то будь-яке число

$\{q_n\} = \{1, 2, \dots, p-1\}$ може бути подане у виді степеня первісного кореня g за модулем p : $q_n \equiv |g^{i_n}|_p$, де $\{i_n\} = \{0, 1, \dots, p-2\}$.

Отже, якщо $A = |g^{i_A}|_p$ і $B = |g^{i_B}|_p$, то їх добуток за модулем p дорівнює:

$$|A \cdot B|_p = |g^{i_A+i_B}|_{p-1}|_p. \quad (5.4)$$

Приклад. Задано числа $A = 3$, $B = 6$, треба знайти їх добуток за модулем 7.

Первісний корінь для модуля $p = 7$ дорівнює $g = 3$, відповідно, відсортована в порядку зростання таблиця індексів має вигляд (таблиця 5.3).

Таблиця 5.3– Таблиця індексів для модуля 7

$q = g^i _7$	1	2	3	4	5	6
i	0	2	1	4	5	3

Перейдемо від чисел $A = 3$, $B = 6$ до їх індексів, отже, $i_A = 1$, $i_B = 3$.

За формулою (5.4) знаходимо результат множення $|A \cdot B|_7 = |3^{1+3}|_7 = 4$.

Знайти результат множення можна також шляхом використання оберненої таблиці. Для цього необхідно додати індекси $i_A = 1$, $i_B = 3$ за модулем $p-1$ і за таблицею 5.4 зйти добуток. Оскільки $|1+3|_6 = 4$, згідно таблиці 5.4 $|A \cdot B|_7 = 4$.

Таблиця 5.4 – Обернена таблиця індексів для модуля 7

i	0	1	2	3	4	5
$q = g^i _7$	1	3	2	6	4	5

Структурна схема пристрою множення на основі індексного методу наведена на рисунку 5.3.

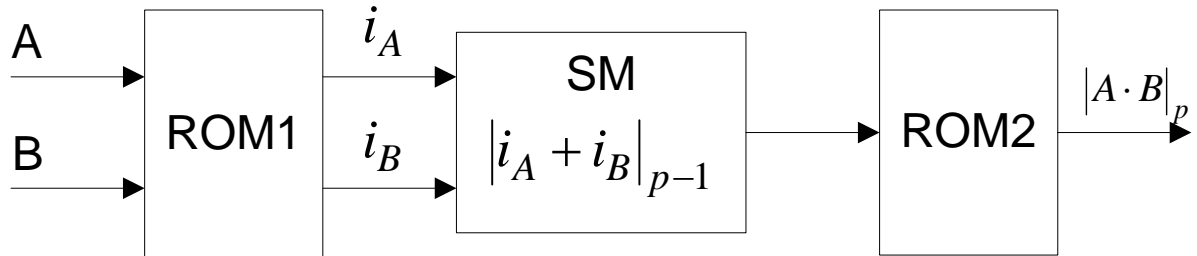


Рисунок 5.3 – Структурна схема пристрою множення на основі індексного методу: ROM1 – пам’ять для зберігання таблиці залишки – індекси; ROM2 – пам’ять для зберігання оберненої таблиці індекси – залишки; SM – суматор за модулем $p - 1$

Обсяг таблиць залежить від значення модуля і визначається кількістю $2(p - 1)$ комірок.

Підвищити швидкодію виконання операції множення на основі індексного методу можна за рахунок розпаралелювання роботи суматора індексів. Це можливо зробити, якщо модуль $p - 1$ можна розбити на попарно взаємно прості множники $p - 1 = m_1 \cdot m_2 \cdot \dots \cdot m_r$. Тоді операція додавання розбивається на r операцій додавання меншої розрядності. В цьому випадку індекси набудуть вигляду: $(|i|_{m_1}, |i|_{m_2}, \dots, |i|_{m_r})$, а додавання проводитиметься незалежно по кожному елементу вектора (рисунок 5.4) [82, 187, 227].

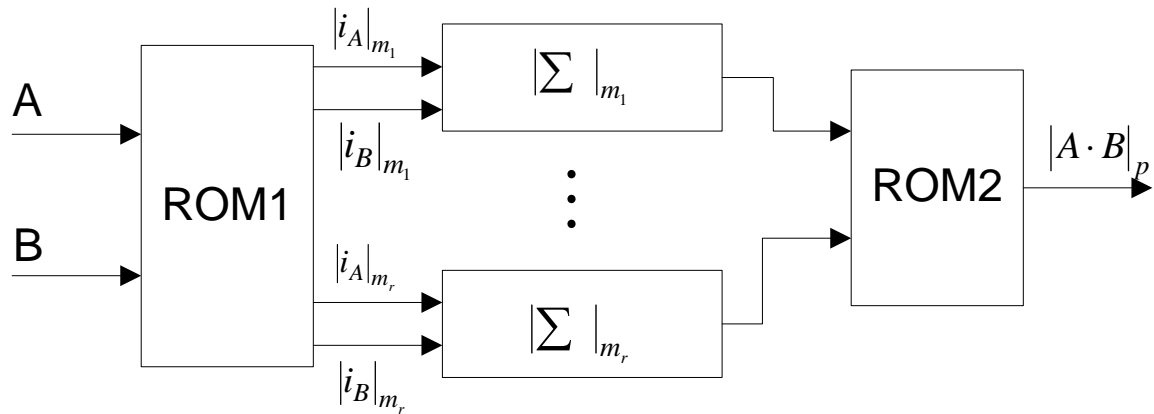


Рисунок 5.4 – Структурна схема пристрою множення на основі індексного методу з розпаралелюванням операції додавання

Наприклад, для модуля 7 (попередній приклад) суматор за модулем 6 можна замінити двома суматорами за модулем 2 і за модулем 3.

При цьому, в пам'яті ROM1 зберігаються залишки від індексів за відповідними модулями (m_1, m_2, \dots, m_r) .

5. Метод модулярного множення на основі різниці квадратів

Для реалізації множення чисел можна використати формулу [187]:

$$A \times B = \frac{(A+B)^2}{4} - \frac{(A-B)^2}{4}. \quad (5.5)$$

Для виконання операції множення за модулем p формулу (5.5) запишемо у вигляді:

$$|A \times B|_p = \left| \frac{(A+B)^2}{4} - \frac{(A-B)^2}{4} \right|_p. \quad (5.6)$$

Операцію ділення на 4 замінимо операцією множення на обернений елемент:

$$|A \times B|_p = \left| \left| (A + B)^2 \cdot |4^{-1}|_p \right|_p - \left| (A - B)^2 \cdot |4^{-1}|_p \right|_p \right|_p,$$

де $|4^{-1}|_p$ – обернений елемент до $|4|_p$.

Тобто, необхідно знайти таке значення x , щоб $|x \cdot 4|_p \equiv 1$ або $|4 \cdot 4^{-1}|_p \equiv 1$. Наприклад, обернений елемент до 4 за модулем $p = 7$ дорівнює 2, оскільки $|4 \cdot 2|_7 \equiv 1$.

На основі формули (5.6) можна синтезувати структурну схему пристрою множення (рисунок 5.5). При описі схеми на мові Verilog блоки 3 і 4 реалізовані у вигляді таблиць, обсяг яких залежить від значення модуля і дорівнює $2 \cdot p - 1$.

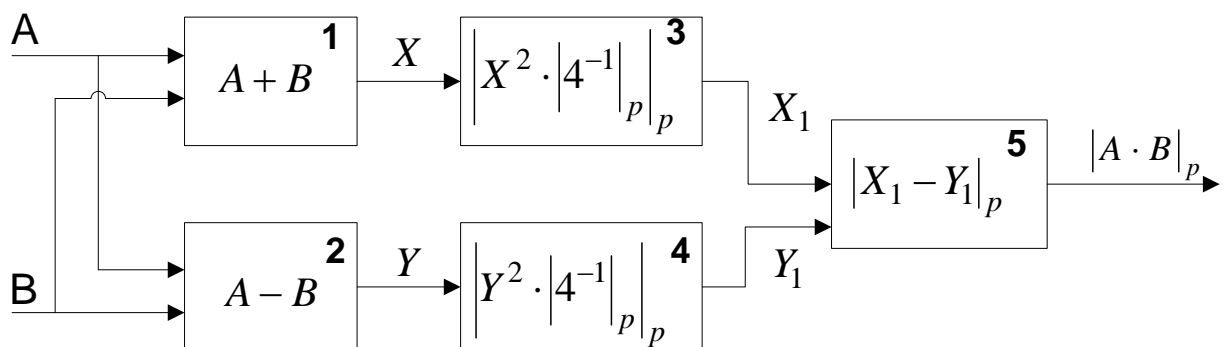


Рисунок 5.5 – Структурна схема пристрою множення на основі методу різниці квадратів

Проведені дослідження апаратної складності виконання операції модулярного множення показали, що при розрядності модуля менше 7 біт більш ефективними є метод індексного множення та метод різниці квадратів (рисунок 5.6) [82, 102].

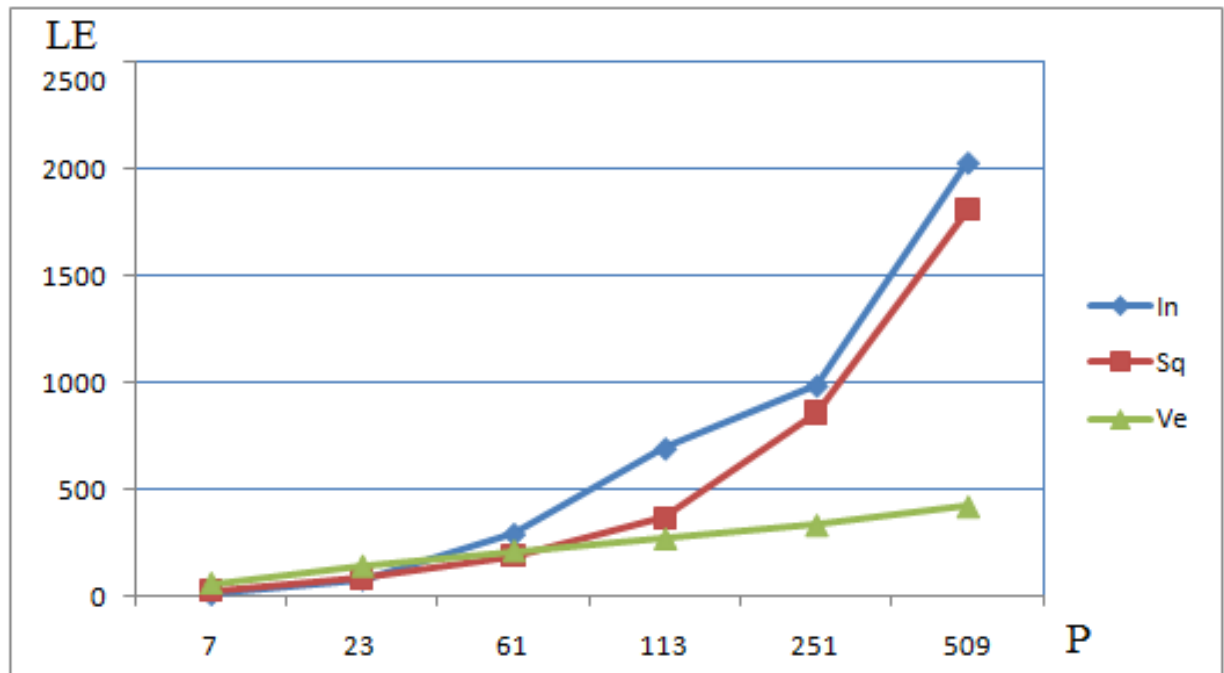


Рисунок 5.6 – Залежність апаратних затрат (кількість логічних елементів LE) від значення модуля при реалізації модулярного множення методами: In – індексне множення, Sq – множення на основі різниці квадратів, Ve – використання операторів мови Verilog.

Однак при зростанні розрядності модулів метод з використанням операторів мови Verilog (див.рисунок 5.6) потребує менше логічних елементів. Дослідження часу затримки виконання операції модулярного множення (рисунок 5.7) показали, що при малій розрядності модулів (3-4 розряди) мінімальний час затримки забезпечує метод індексного множення, а при розрядності 5 біт і більше кращий результат забезпечує метод різниці квадратів.

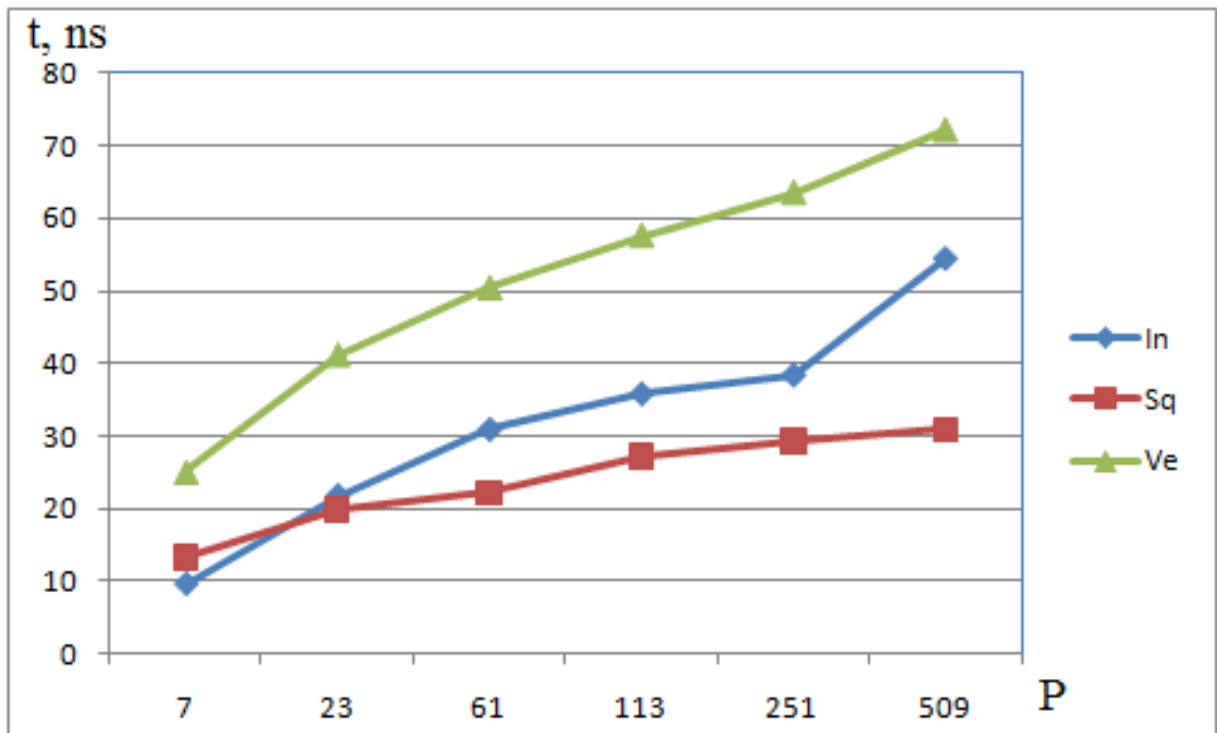


Рисунок 5.7 – Залежність часу затримки від значення модуля при реалізації модулярного множення методами: In – індексне множення, Sq – множення на основі різниці квадратів, Ve – використання операторів мови Verilog.

Експериментально підтверджено, що для дослідження ефективності виконання операції модулярного множення для різних наборів модулів та діапазону подання доцільно вибрати наступні три методи: метод 1 – використання операторів мови Verilog; метод 2 – індексне множення, метод 3 – різниця квадратів [44] та дві серії мікросхем: Max II та Cyclone II (таблиця 5.5).

Таким чином, проведені експериментальні дослідження показали, що найбільш ефективним є метод різниці квадратів. При розрядності модулів (3-5 біт) апаратні затрати в 2 - 3 рази менші, порівняно із апаратними затратами при розрядності модулів 8-10 біт (таблиця 5.5).

Таблиця 5.5 – Параметри виконання операції множення в СЗК

Діапазон даних 2^n та значення модулів	Кількість логічних елементів			Час затримки, нс		
	Метод 1	Метод 2	Метод 3	Метод 1	Метод 2	Метод 3
n=8: 3, 7, 13	188	109	78	33,94	12,28	17,1
n=16 239, 277	757	2574	2719	71,37	35,94	31,23
29, 43, 47	552	758	421	46,61	28,84	23,1
3, 7, 11, 13, 23	430	409	207	40,06	24,33	20,04
n=24: 4091, 4111	1493 (Max II)	*		105,85 (Max II)	–	
	1042 (Cyclone II)			68,28 (Cyclone II)		
229, 239, 307	1089 (Max II)	4015 (Cyclone II)	4342 (Cyclone II)	74,91 (Max II)	40,18 (Cyclone II)	33,96 (Cyclone II)
	731 (Cyclone II)			46,29 (Cyclone II)		
7, 23, 29, 59, 61	767 (Max II)	883 (Max II)	596 (Max II)	48,06 (Max II)	29,25 (Max II)	22,7 (Max II)
	516 (Cyclone II)	900 (Cyclone II)	596 (Cyclone II)	32,76 (Cyclone II)	25,28 (Cyclone II)	20,1 (Cyclone II)

* – недостатньо апаратних ресурсів у вибраних серіях мікросхем.

Використання методу різниці квадратів (метод 3) підвищує швидкодію виконання операції множення в середньому на 40%. Ефективність виконання операції множення, зокрема, апаратні затрати, також залежить від значення вибраних модулів та їх кількості.

Отримані результати досліджень будуть використані при розробці пристроїв обробки даних на основі модулярних коректуючих кодів [102], описаних в пунктах 3.1, 3.2.

5.2 Пристрій виявлення та виправлення однократних помилок на основі модулярних коректуючих кодів

Для зменшення навантаження на безпроводний контролер, зокрема зменшення часу затримки обробки даних, при застосуванні модулярних коректуючих кодів кодер/декодер реалізовано на ПЛІС.

5.2.1 Синтез кодера

Процес обробки полягає в обчислення перевірного символу, яке здійснюється за формулою (3.1). Для реалізації паралельного виконання операцій множення та отримання залишку рівняння (3.1) запишемо у вигляді [152, 213]:

$$\bar{x}_{k+1} = \left| \left| v_1 \cdot x_1 \right|_p + \left| v_2 \cdot x_2 \right|_p + \dots + \left| v_i \cdot x_i \right|_p + \dots + \left| v_k \cdot x_k \right|_p \right|_p. \quad (5.6)$$

На основі формули (5.6) синтезовано узагальнену структуру кодера, який складається з блоків модулярного множення та блоку додавання за модулем (рисунок 5.8). Кількість блоків множення дорівнює кількості інформаційних символів в коректуючому коді.

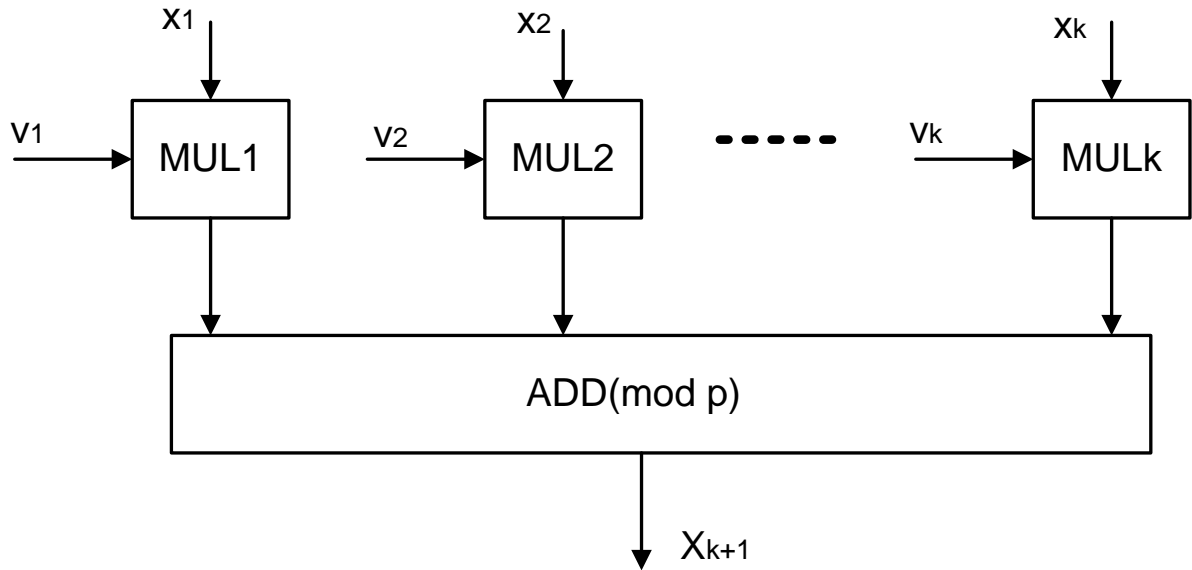


Рисунок 5.8 – Узагальнена структура кодера: MUL_i – блоки множення за модулем p ; ADD – блок додавання по модулю p .

На основі узагальненої структури (див. рисунок 5.8) розроблено функціональну схему кодера (рисунок 5.9) [152, 213].

Схема кодера описана на мові Verilog в середовищі Quartus II та реалізована на ПЛІС фірми «Altera» серії MAX II, мікросхема EPM240T100C3 (рисунок 5.10).

На виході $out[9..0]$ формуються перевіірочні символи в залежності від даних на входах in_1 - in_9 (рисунок 5.9). Часові діаграми роботи кодера наведені на рисунку 5.11.

Апаратні затрати на реалізацію кодера становлять: загальна кількість логічних елементів 58 / 240 (24%) (таблиця 5.6), максимальна затримка формування коректуючого символу становить 43,7 нс. Експериментально встановлено, що реалізація кодера на основі рівняння (5.6) зменшує затримку формування коректуючого символу приблизно на 30 % за рахунок паралельної роботи блоків множення по модулю p з використанням методу індексного множення.

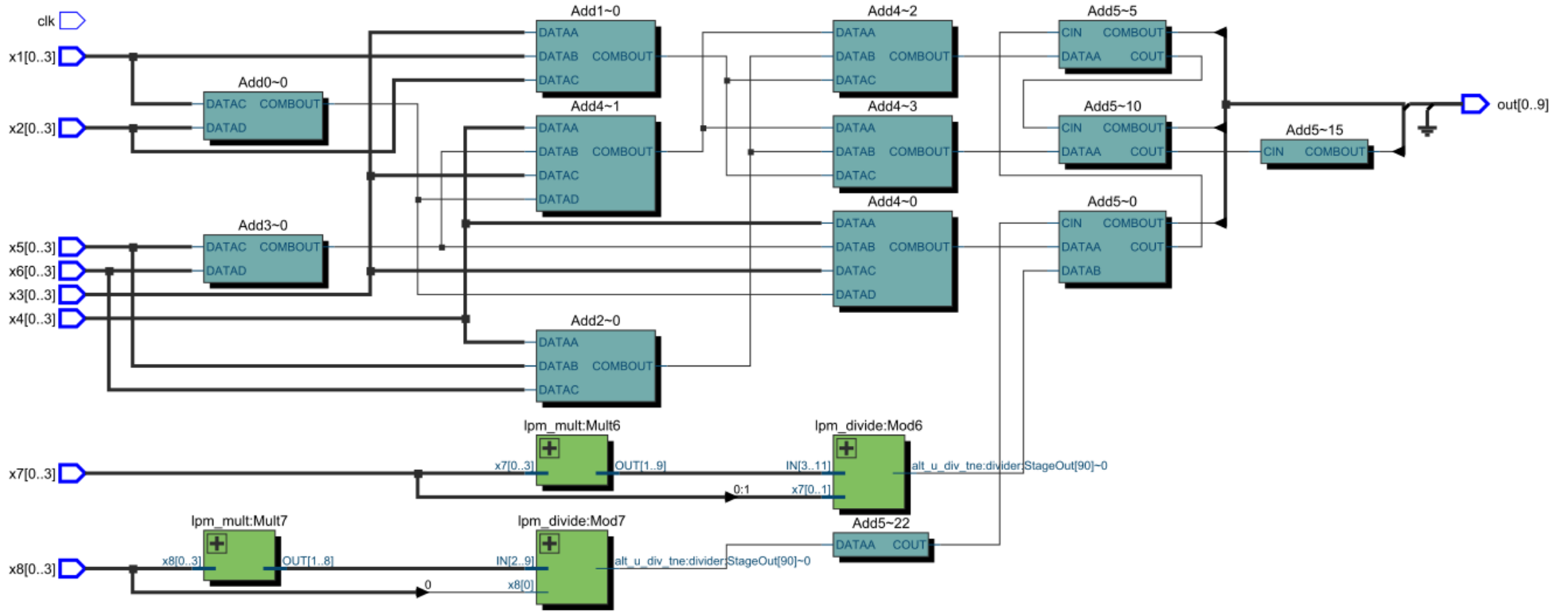
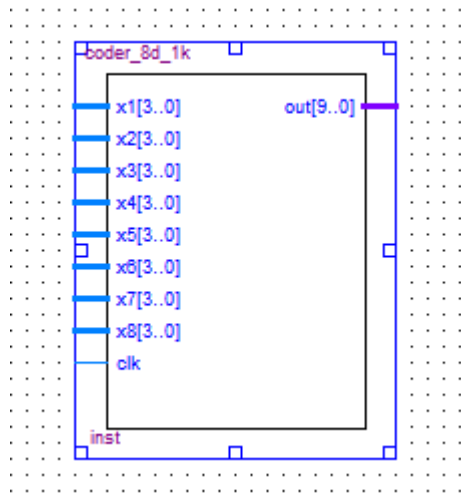


Рисунок 5.9 – Функціональна схема кодера



Ім'я	Опис	Розрядність
x1 – x8	Входи	4 біти
out	Вихід	10 біт
clk	Тактовий вхід	1 біт

Рисунок 5.10 – Інтерфейс кодера

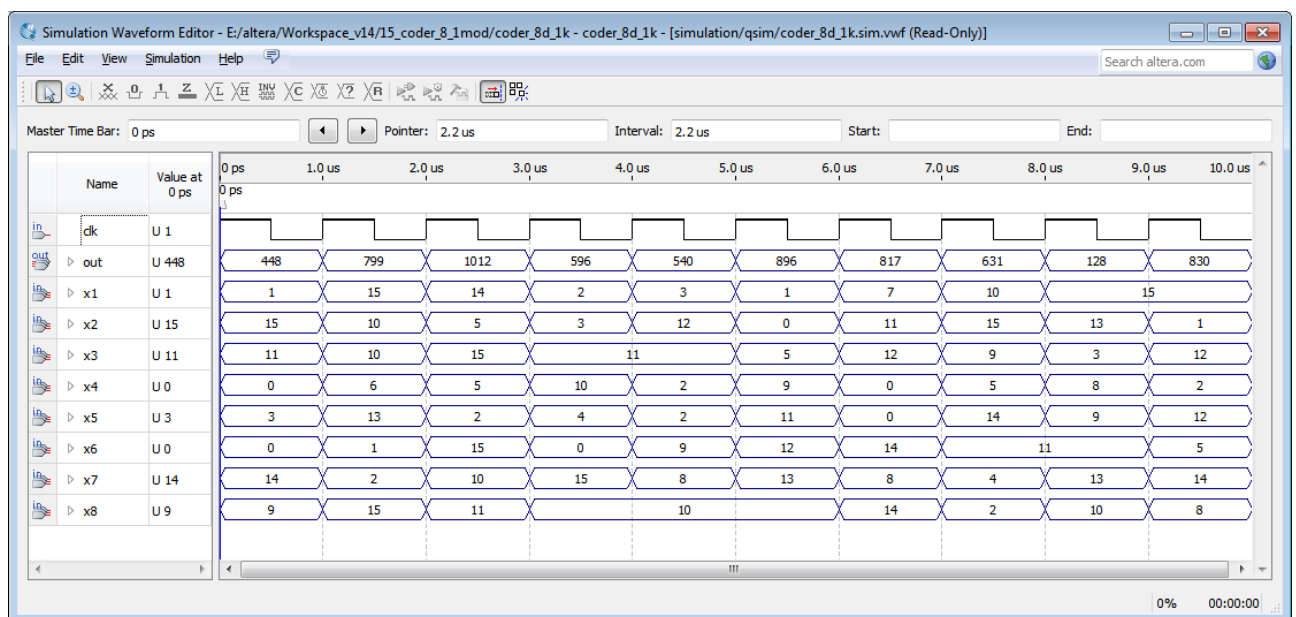


Рисунок 5.11 – Результати симуляції роботи кодера

Таблиця 5.6 – Параметри синтезу кодера

Назва	Значення
Назва файлу топ-рівня	coder
Серія ПЛІС	MAX II
Мікросхема	EPM240T100C3
Всього логічних елементів	58/240 (24%)
UFM блоків	0 / 1(0%)
Всього контактів	43/80 (54%)
Час затримки, нс	43,736

5.2.2 Синтез декодера

На основі запропонованого в пункті 2.2 алгоритму декодування модулярних коректуючи кодів розроблено узагальнену структура декодера виправлення однократних помилок (рисунок 5.12) [219].

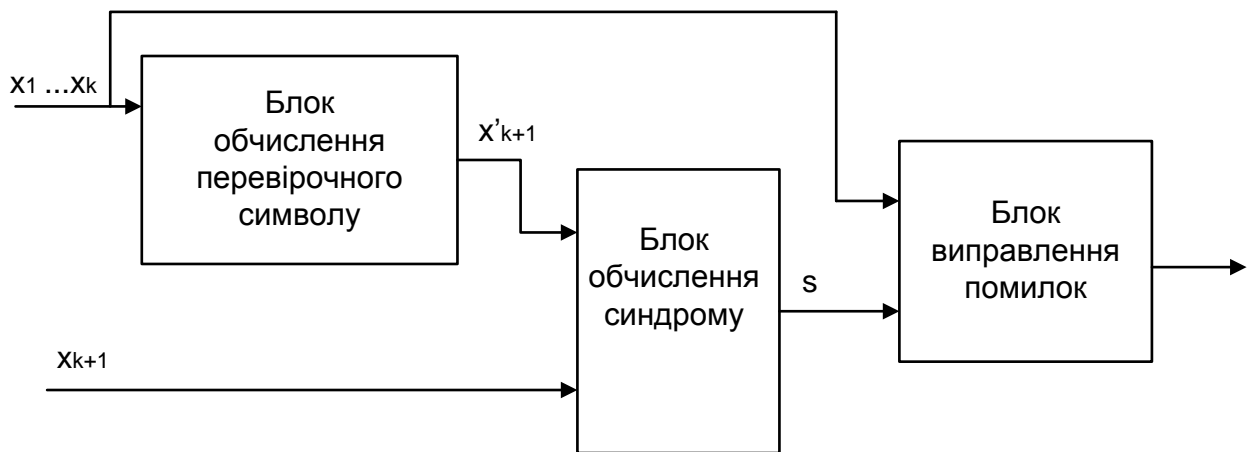


Рисунок 5.12 – Узагальнена структура декодера

Декодер складається з трьох блоків: блоку обчислення перевірочних символів по прийнятих даних, блоку обчислення синдрому та блоку виправлення помилок.

Робота декодера базується на наступних кроках:

- 1) обчислення перевірного символу по прийнятих даних;
- 2) виявлення помилки;
- 3) виправлення помилки.

Перший крок полягає в обчисленні перевірного символу по прийнятих даних на основі формули (5.6). Виявлення помилки відбувається шляхом обчислення синдрому (порівняння обчисленого та прийнятого перевірного символу). Якщо синдром не дорівнює нулю, помилка виявлена і сигнал поступає на блок виправлення помилок.

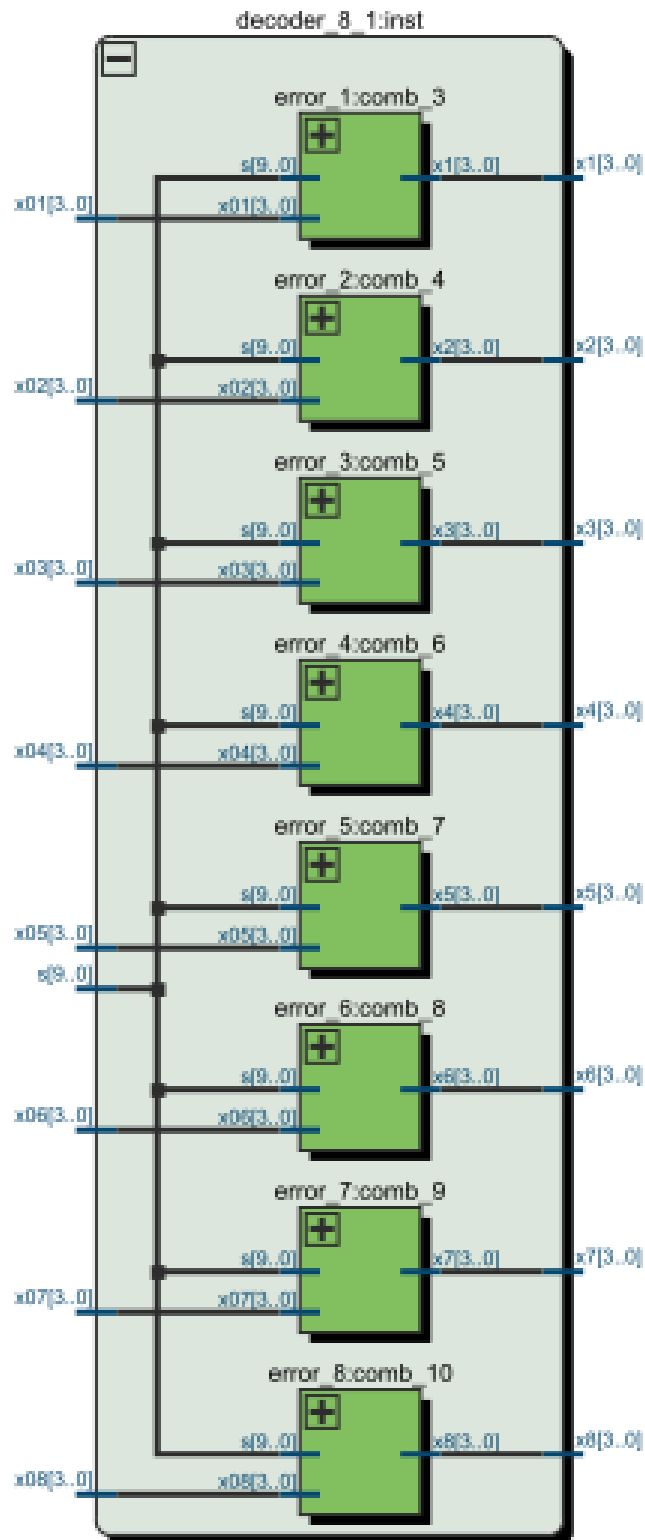


Рисунок 5.13 – Функціональна схема блоку виправлення помилок

Третій крок полягає в порівнянні значення синдрому з попередньо обчисленими значеннями синдрому при всіх можливих варіантах помилки для кожного блоку даних (див. рисунок 5.13). На вхід блоку виправлення

помилкок поступають вхідні сигнали та значення синдрому. В блоках error1, ..., error_8 записані таблиці, на вхід яких поступають значення синдрому, а на виході отримуємо виправлені значення інформаційних символів.

На основі узагальненої структури (див. рисунок 5.12) розроблено функціональну (рисунок 5.14) та принципову (рисунок 5.15) схеми декодера відповідно. Результати синтезу декодера наведені в таблиці 5.7.

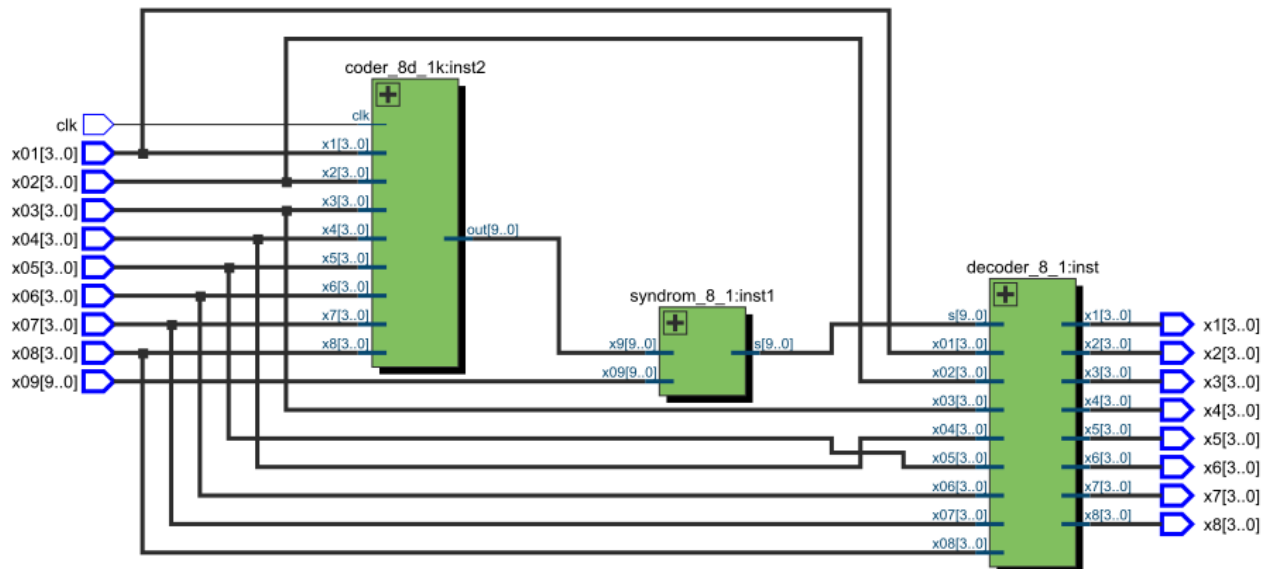


Рисунок 5.14 – Функціональна схема декодера

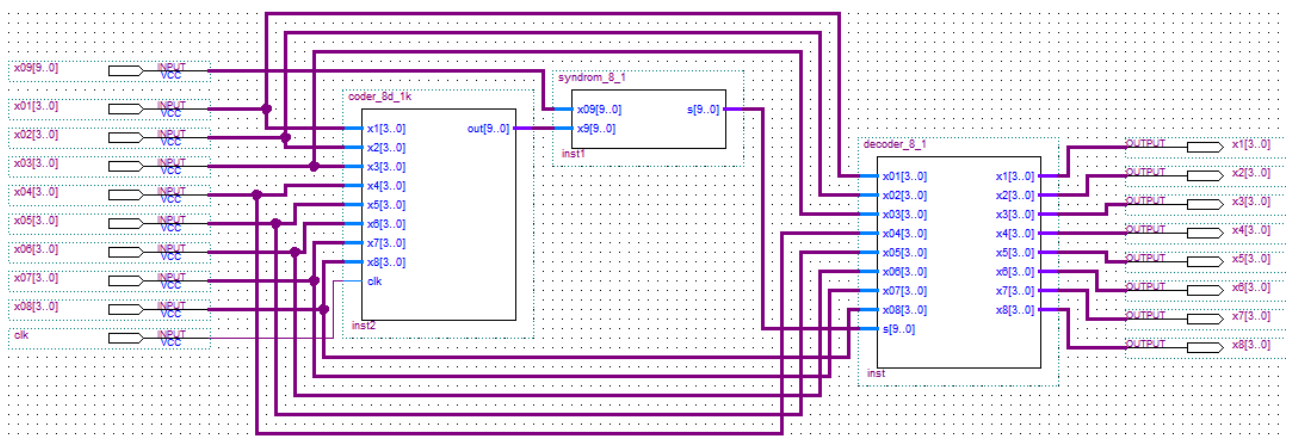


Рисунок 5.15 – Принципова схема декодера

Таблиця 5.7 – Параметри синтезу декодера

Назва	Значення
Назва файлу топ-рівня	decoder
Серія ПЛІС	MAX II
Мікросхема	ЕРМ1270Т144С3
Всього логічних елементів	1108/1270 (87%)
UFM блоків	0 /1(0%)
Всього контактів	75/116 (65%)
Всього віртуальних контактів	0
Час затримки, нс	62,634

Декодер для коду з вісьмома блоками даних і одним контрольним символом реалізований на ПЛІС фірми Altera, серії MAX II, Device ЕРМ1270Т144С3. Апаратні затрати: total logic elements 1108 / 1270 (87%), максимальна затримка формування вихідних сигналів становить 62,6 нс (таблиця 5.7).

Для перевірки роботи кодера/декодера розроблено модуль введення помилок хог_4_8b (рисунок 5.16), який дозволяє вводити помилки в будь-який з блоків даних. Для імітації спотворення даних необхідно на вхід e_i відповідного каналу, ввести вектор помилки. Одиничний біт у векторі помилки інвертує відповідний розряд в блоці даних.

В процесі симуляції роботи кодера/декодера (рисунок 5.17) для компактнішого подання значень, тактовий сигнал наведений у двійковій системі, а всі інші значення – у десятковій системі числення. Так, при векторі помилки $e_2 = 0101$ на вході ЕО2 відбулося спотворення даних в блоці b_2 , $b_2' = b_2 \oplus e = 0100 \oplus 0101 = 0001$, значення синдрому $s_{2i} = 970 \oplus$ – додавання по модулю 2. Отже, згідно додатку В, синдром $s_{2i} = 970$

відповідає значенню помилки $x_i = 3$, тобто для виправлення помилки необхідно $b2 = b2' + x = 1 + 3 = 4$.

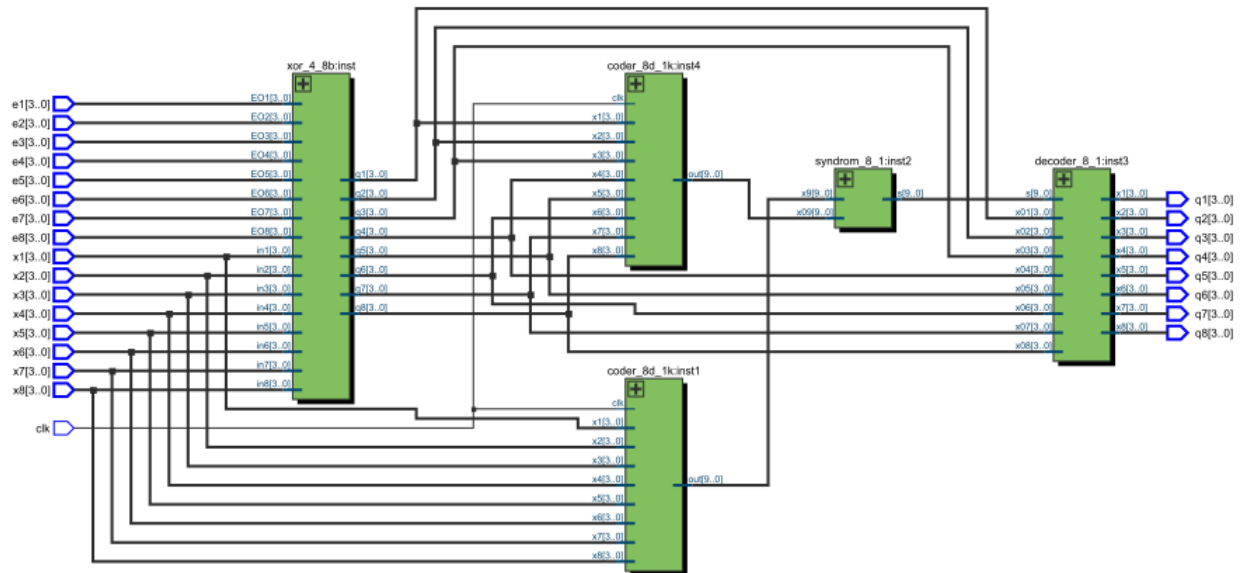


Рисунок 5.16 – Функціональна схема верифікації роботи кодера/декодера модулярного коректуючого коду: $e1 - e8$ – входи вводу помилок; $x1 - x8$ – інформаційні входи; $q1 - q8$ – інформаційні виходи.

Результати верифікації розроблених пристроїв підтвердили їх коректну роботу, що ілюструється часовою діаграмою, на якій показано виправлення помилки на виході $q2$ (при векторі помилки $e2=5$) (рисунок 5.17).

Реалізовані на ПЛІС кодер та декодер модулярного коректуючого коду забезпечують виправлення однократних помилок (помилки в одному блоці даних). Даний принцип може бути використаний для реалізації кодера та декодера з іншими характеристиками коду, для цього необхідно змінити кількість вхідних блоків даних та значення коефіцієнтів і контрольного модуля в описі відповідного блоку на мові Verilog.

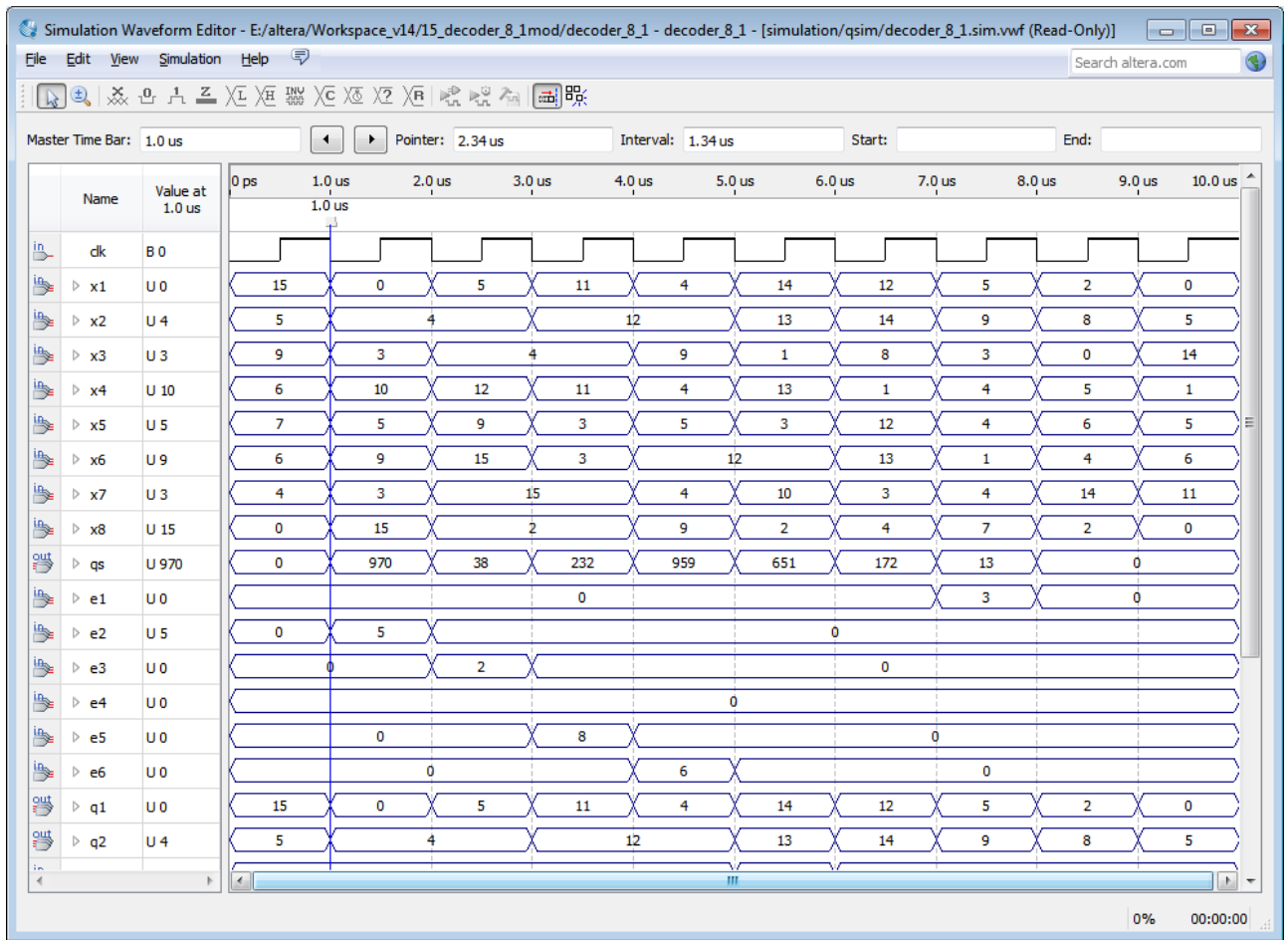


Рисунок 5.17 – Результати симуляції роботи декодера

Робота кодера і декодера описана на мові Verilog в середовищі Quartus II та реалізована на ПЛІС фірми Altera, серії MAX II. Апаратні затрати кодера: total logic elements – 58, максимальна затримка формування коректуючого символу становить 42,74 нс. Апаратні затрати декодера: total logic elements – 1108, максимальна затримка формування вихідних сигналів становить 62,6 нс.

Таким чином, в підрозділі 5.2 синтезовано узагальнені схеми, розроблено функціональні структури та реалізовано на ПЛІС кодер/декодер, який виявляє та виправляє однократні помилки на основі модулярного коректуючого коду. Враховуючи низькі апаратні затрати на реалізацію кодера/декодера модулярного коректуючого коду та низьке енергоспоживання сучасних ПЛІС, розроблені пристрої будуть використані

для підвищення надійності передачі даних в безпроводних сенсорних мережах системи контролю електричних підстанцій.

5.3 Синтез пристрою виправлення помилок у двох символах на основі модулярних коректуючих кодів

На основі теоретичних основ модулярних коректуючих кодів, поданих в розділі 3, розроблено пристрій, який забезпечує виявлення та виправлення помилок в двох інформаційних символах [219]. Даний пристрій складається з блоків кодування та декодування. Виправлення помилок реалізовано в декодері (рисунок 5.18). Вхідні дані X^i поступають на блок вхідних регістрів, з виходу яких інформаційні символи подаються на блок обчислення перевірочних символів і блок виправлення помилок, а перевірочні – на блок обчислення синдрому.

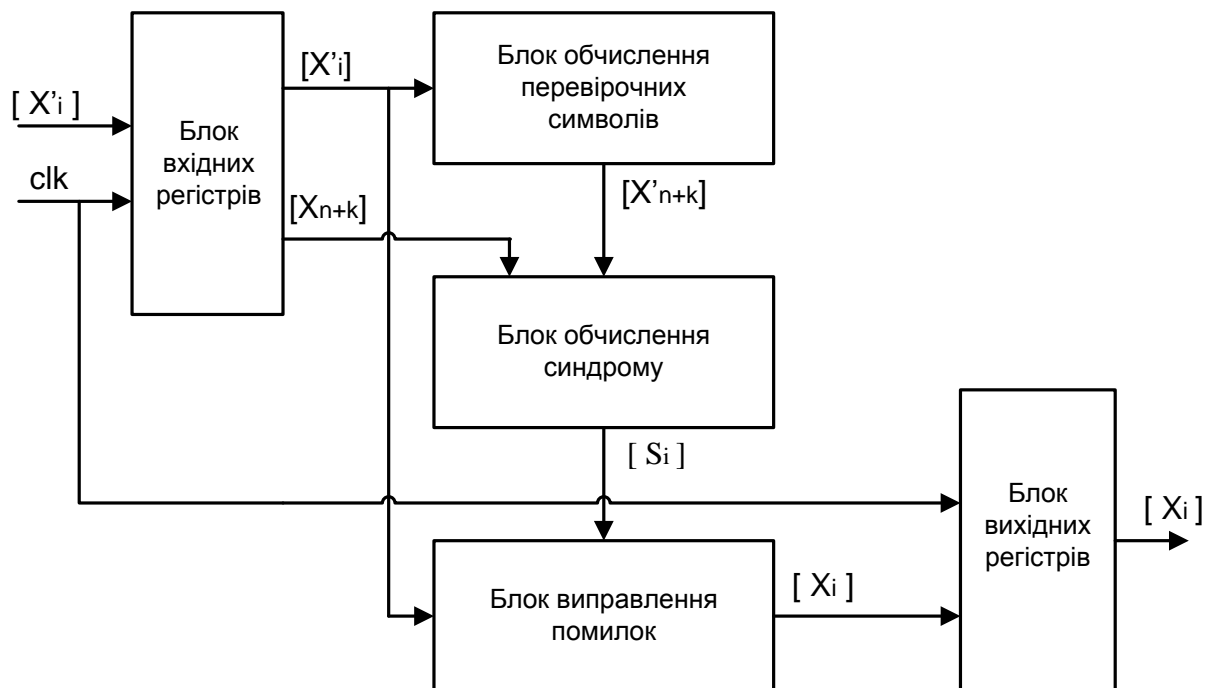


Рисунок 5.18 – Структурна схема пристрою виправлення помилок

Обчислене значення синдрому (на основі отриманих перевірочних символів X^{n+k}) поступає на блок виправлення помилок. Виправлені дані надходять на блок вихідних регістрів.

Розроблено алгоритм декодування даних (рисунок 5.19), де в блоці 3 на основі обчислення синдрому виявляються помилки. При виявленні помилки її локалізація та виправлення відбувається в блоці 5 за алгоритмом, наведеним на рисунку 5.20. В блоках 2-4 (рисунок 5.20) реалізовано цикл, за допомогою якого здійснюється виявлення та виправлення помилок в двох інформаційних символах. В блоці 5 відбувається перевірка: якщо знайдені значення інформаційних символів (x_i, x_j) знаходяться в робочому діапазоні N , то помилки виправлені.

Для реалізації декодера необхідно дослідити блок обчислення перевірочних символів, блок обчислення синдрому та блок виправлення помилок.

Розглянемо детально функціонування блоку виправлення помилок. Виправлення помилок в двох інформаційних символах зводиться до розв'язання системи рівнянь [219]:

$$|v_i \cdot e_i + v_j \cdot e_j|_P = \delta_1, \quad (5.7)$$

$$|w_i \cdot e_i + w_j \cdot e_j|_P = \delta_2, \quad (5.8)$$

Для розв'язку рівнянь (5.7,5.8) запропоновано наступний алгоритм [61].

- 1) Множимо рівняння (5.7) на коефіцієнт w_i , а рівняння (5.8) на v_i .
- 2) Віднімемо від рівняння (5.7) рівняння (5.8), в результаті отримаємо рівняння:

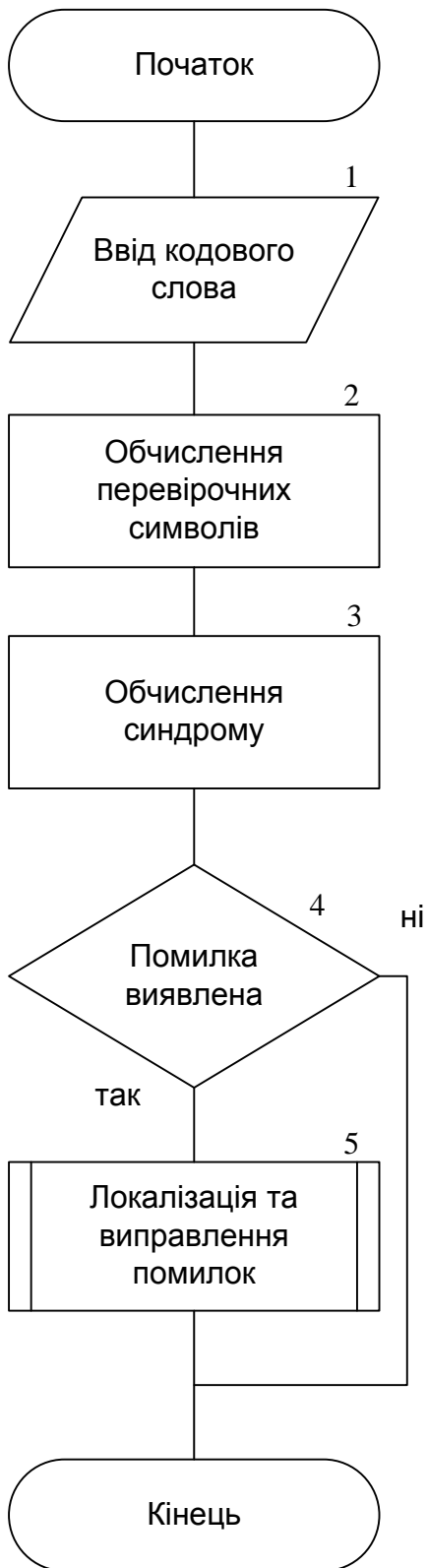


Рисунок 5.19 – Блок – схема алгоритму декодування даних

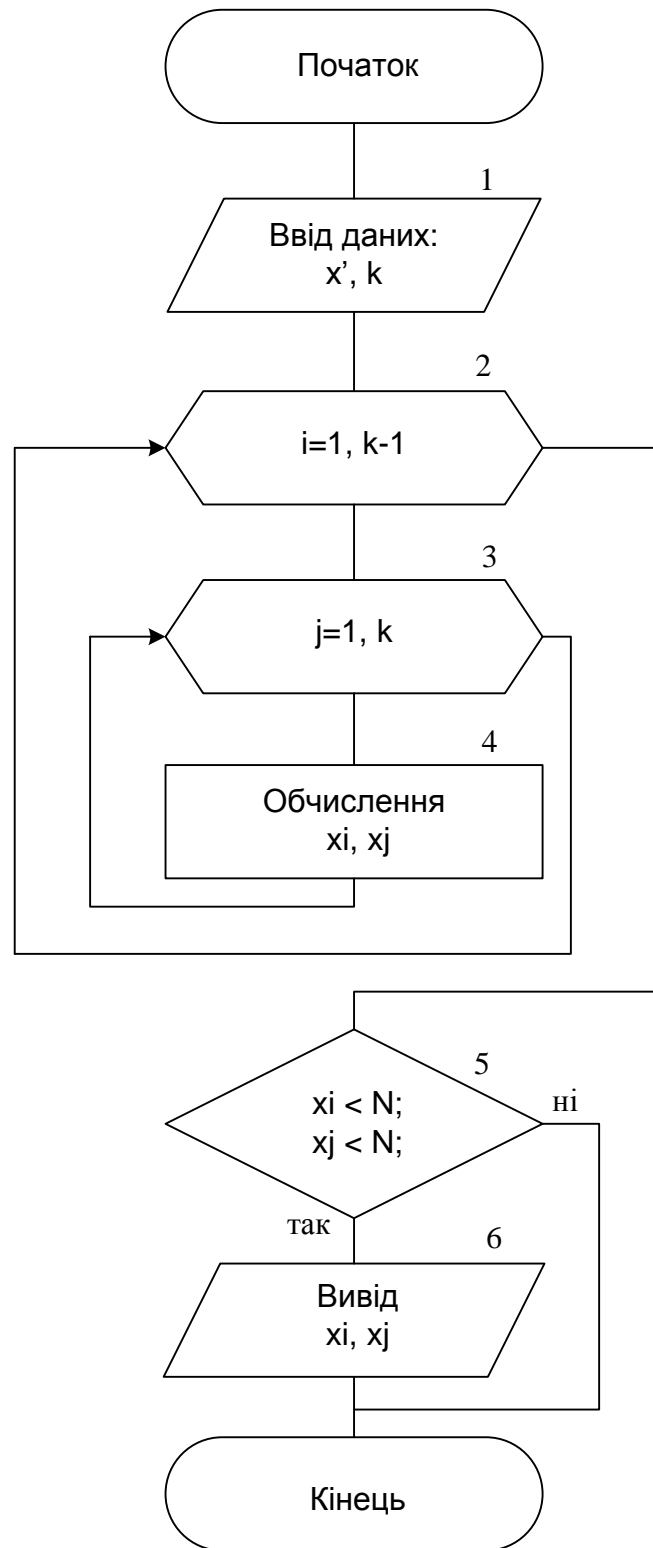


Рисунок 5.20 – Блок – схема функції локалізації та виправлення помилок

$$|u \cdot e_2|_p = y. \quad (5.9)$$

- 3) Знаходимо обернений елемент до u по модулю p , $u^{-1}(\text{mod } p)$.
- 4) Помноживши ліву і праву частини рівняння (5.9) на обернений елемент, отримаємо:

$$|u^{-1} \cdot u \cdot e_2|_p = |y \cdot u^{-1}|_p,$$

$$|u^{-1} \cdot u|_p = 1,$$

$$e_2 = y_0,$$

де $y_0 = y \cdot u^{-1}$.

- 5) Підставивши знайдене значення e_2 в рівняння (5.8) і виконавши операції за пунктами 3 і 4, знаходимо e_1 .

Обчислення оберненого елемента. Цю процедуру доцільно реалізувати на ПЛІС у вигляді окремого блоку [219]. Для обчислення мультиплікативного оберненого елемента по модулю $u^{-1}(\text{mod } p)$ використаємо розширений алгоритм Евкліда [1, 227]. Оскільки модуль p просте число, то $\text{gcd}(a, p) = 1$, тому отримаємо

$$a \cdot x + p \cdot y = 1,$$

$$a \cdot x = 1 - p \cdot y,$$

звідки

$$a \cdot x \equiv 1(\text{mod } p),$$

де x – мультиплікативний обернений елемент до a по модулю p .

Алгоритм обчислення мультиплікативного оберненого елемента за розширеним алгоритмом Евкліда складається з наступних кроків:

1. Вводимо значення: $a_0 = p$, $a_1 = a$, $x_0 = 0$, $x_1 = 1$.
2. Обчислюємо частку q і залишок r за формулами:

$$q = \left[\frac{a_0}{a_1} \right],$$

$$r = a_0 \% a_1.$$

3. Якщо $r \neq 0$, тоді оновлюємо змінні: $a_0 = a_1$, $a_1 = r$, $t = x_1$,
 $x_1 = x_0 - q \cdot x_1$, $x_0 = t$ і повертаємося до кроку 2. Інакше крок 4.
4. Якщо $x_1 < 0$, тоді $x_1 = x_1 + p$, $u^{-1} = x_1$. Інакше $u^{-1} = x_1$.

Реалізація обчислення мультиплікативного оберненого елемента на мові Verilog має вигляд:

```

module evclid_1 (a, p, x0, x1, clk);
input clk;
input [9:0]a, p;
output reg [9:0]x0, x1;
reg [9:0]q, r, a0, a1, t;
always @(posedge clk)
begin
x0 = 0;
x1 = 1;
while( a1 != 0 )
begin
q = a0 / a1;
r = a0 % a1;
a0 = a1;
a1 = r;
t = x1;
x1 = x0 - q * x1;
x0 = t;
end
end
endmodule

```


Результати обчислення оберненого елемента наведені на рисунку 5.21. Наприклад, обернений елемент для $u = 97$ за модулем $p = 1021$ дорівнює $u^{-1} = 200$ (рисунок 5.21).

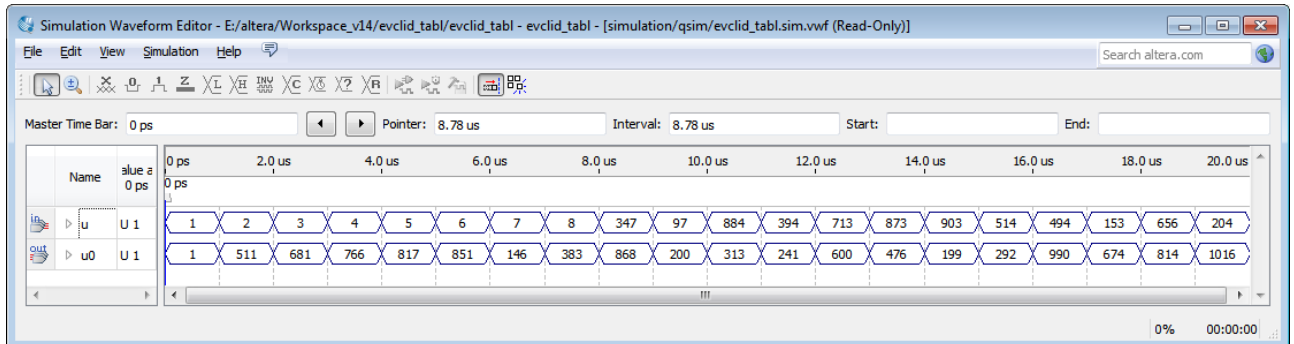


Рисунок 5.21 – Результати моделювання блоку обчислення оберненого елемента

Для локалізації помилок в будь-яких двох інформаційних символах виразимо помилки e_i , e_j через відомі значення (синдром і взаємно прості коефіцієнти):

$$e_j = \left| (s_2 \cdot v_i - s_1 \cdot w_i) \cdot u_{1i}^{-1} \right|_p ;$$

$$e_i = \left| (s_2 - (w_j \cdot e_j)) \cdot u_{2j}^{-1} \right|_p ;$$

де $u_{1i} = \left| v_i \cdot w_j - v_j \cdot w_i \right|_p$, $u_j = w_i$;

s_1 , s_2 – синдром;

v_i , v_j , w_i , w_j – взаємно прості коефіцієнти.

На основі розробленої структурної схеми декодера (див. рисунок 5.18) та алгоритмів роботи декодера (див. рисунок 5.19, 5.20) синтезовано функціональну схему декодера (рисунок 5.22). Останній складається з наступних функціональних блоків:

calculation_u – обчислення коефіцієнта u при e_2 ;

- evclid_tablcase – обчислення оберненого елемента до u , u^{-1} ;
 coder_2 – обчислення перевірочних символів;
 syndrome_8_2 – обчислення синдрому;
 correct_e1 – обчислення помилки в першому інформаційному символі e_1 ;
 correct_e2 – обчислення помилки в другому інформаційному символі e_2 ;
 correct_end – виправлення помилок в інформаційних символах.

Назви та призначення входів/виходів декодера наведені в таблиці 5.8.

Verilog-код декодера наведений в додатку Д.

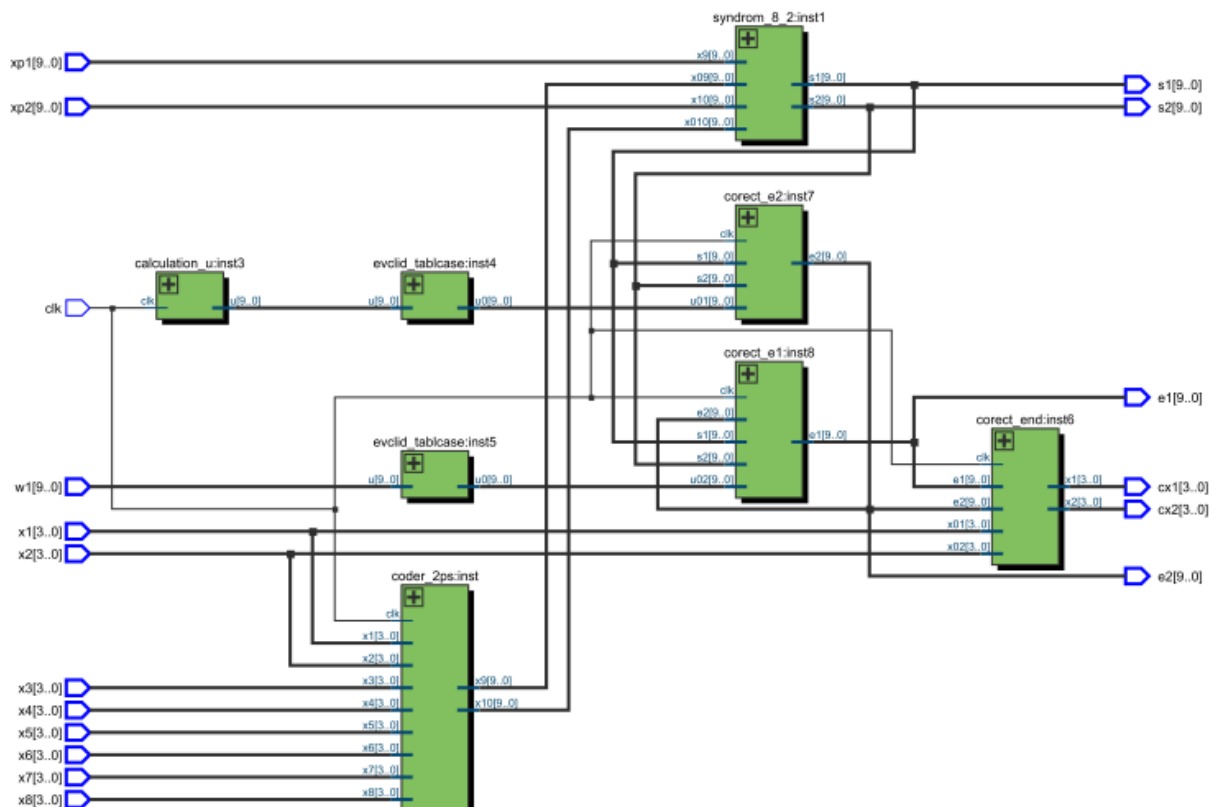


Рисунок 5.22 – Функціональна схема декодера

Принципова схема декодера приведена в додатку Г. Проведена оцінка апаратних затрат при використанні різних серій ПЛІС фірми Altera (таблиця 5.9) показала, що для реалізації декодера на мікросхемі EP4CGX22CF19C6, серії Cyclone IV GX без використання вбудованих блоків необхідно 6986 логічних елементів (рисунок 5.22).

Таблиця 5.8 – Назви та призначення входів/виходів декодера

Назва	Тип	Призначення
x1[3..0] – x8[3..0]	input	Інформаційні символи: x1– x8
xp1[9..0]	input	Перевірочний символ
xp2[9..0]	input	Перевірочний символ
clk	input	Тактовий вхід
w1[9..0]	input	Коефіцієнт
s1[9..0]	output	Значення синдрому, s1
s2[9..0]	output	Значення синдрому, s2
e1[9..0]	output	Значення помилки, e1
e2[9..0]	output	Значення помилки, e2
cx1[3..0]	output	Виправлений інформаційний символ, x1
cx2[3..0]	output	Виправлений інформаційний символ, x2

Результати моделювання роботи декодера підтвердили коректність його функціонування (рисунок 5.23)

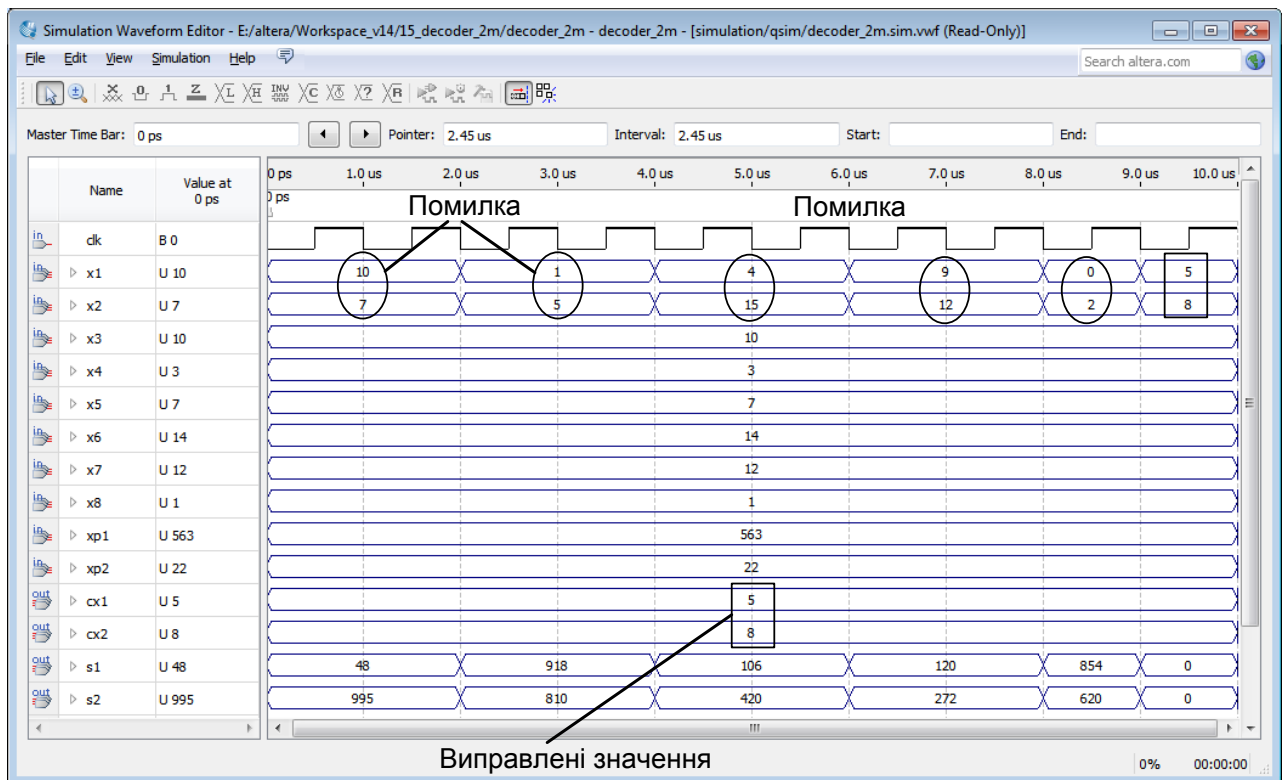


Рисунок 5.23 – Результати моделювання роботи декодера

Таблиця 5.9 – Апаратні затрати на реалізацію декодера

Серія ПЛІС	Загальна кількість логічних елементів	Використання вбудованих блоків
Arria II GX	4938/36100(14%)	Блоки DSP: 16/232(63%)
Cyclone IV E	6034/10320(58%)	Перемножувачі 9 бітних чисел: 16/46(35%)
Cyclone IV GX	6986/21280 (33 %)	–
Cyclone V	2761/56480(5%)	Блоки DSP: 29/156(19%)

Таким чином, проведена оцінка апаратних затрат при використанні різних серій ПЛІС фірми “Altera” показала, що реалізація декодера на мікросхемі EP4CGX22CF19C6, серії Cyclone IV GX без використання вбудованих блоків займає 6986 логічних елементів.

5.4 Пристрій багаторівневого перетворення даних з системи залишкових класів у позиційний код

Для забезпечення можливості апаратної реалізації методу багаторівневого перетворення даних у СЗК розглянемо приклад збору даних з сенсорів температури. Роздільна здатність температурного перетворювача DS18B20 може бути змінена користувачем і становить 9 – 12 бітів, що відповідає дискретності вимірювання температури: 0.5 °С, 0.25 °С, 0.125° С, 0.0625 °С, відповідно. Формат подання температури наведений на рисунку 5.24.

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
MS Byte	S	S	S	S	S	2^6	2^5	2^4

Рисунок 5.24 – Формат регістру температури

Оскільки максимальна роздільна здатність температурного перетворювача становить 12 біт, відповідно вибираємо систему взаємно простих модулів СЗК із умови $p_i > 2^{12}$, для 16-ти вхідних каналів: $p_1 = 4099$, $p_2 = 4111$, $p_3 = 4127$, $p_4 = 4129$, $p_5 = 4133$, $p_6 = 4139$, $p_7 = 4153$, $p_8 = 4157$, $p_9 = 4159$, $p_{10} = 4177$, $p_{11} = 4201$, $p_{12} = 4211$, $p_{13} = 4217$, $p_{14} = 4219$, $p_{15} = 4229$, $p_{16} = 4231$.

Враховуючи велику розрядність вихідних даних (193 біти), для реалізації пристрою перетворення вхідних даних сенсорів, поданих в СЗК, в двійкову систему числення використано метод оснований на китайській теоремі про залишки [210] та паралельно-послідовному відновлення позиційного подання числа за двома залишками. Перевагою даного методу є менша розрядність даних в процесі виконання проміжних обчислень, що в свою чергу забезпечує меншу апаратну складність.

Відновлення позиційного подання числа X за двома залишками [1, 210]:

$$X = x_2 + \left| \left| x_1 - x_2 \right|_{p_1} \cdot k \right|_{p_1} \cdot p_2, \quad (5.10)$$

де x_1, x_2 – залишки по модулях p_1, p_2 ;

k – обернений елемент до p_2 по модулю p_1 , $k = \left| p_2^{-1} \right|_{p_1}$.

На основі співвідношення (5.10) синтезовано структурну схему блоку відновлення позиційного подання числа (рисунок 5. 25).

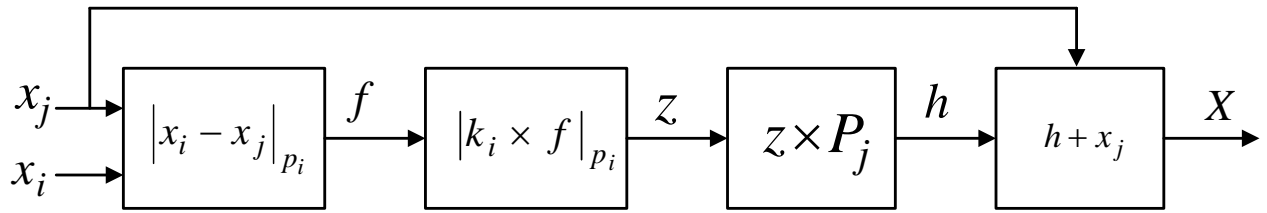


Рисунок 5.25 – Структурна схема блоку відновлення позиційного подання числа за двома залишками

Обробка даних сенсорів відбувається за наступним алгоритмом [217, 220]:

- 1) вхідні дані сенсорів ($n = 16$) об'єднуюємо по два;
- 2) відновлюємо позиційне подання чисел за двома залишками за формулою (5.10);
- 3) повторюємо пункт 2 для всіх пар вхідних даних.
- 4) вихідні дані з пункту 3 групуємо по два.
- 5) відновлюємо позиційне подання чисел за двома залишками і т.д.

В результаті отримаємо $\log_2 n$ рівнів. Для наведеного прикладу при $n = 16$ отримаємо чотири рівні, оскільки $\log_2 16 = 4$.

Об'єднання залишків за двома модулями на кожному рівні відбувається за формулами:

- перший рівень:

$$X_{11} = x_2 + \left| |x_1 - x_2|_{p_1} \cdot k_1 \right|_{p_1} \cdot p_2,$$

$$X_{12} = x_4 + \left| |x_3 - x_4|_{p_3} \cdot k_2 \right|_{p_3} \cdot p_4,$$

$$X_{13} = x_6 + \left| |x_5 - x_6|_{p_5} \cdot k_3 \right|_{p_5} \cdot p_6,$$

$$X_{14} = x_8 + \left| x_7 - x_8 \right|_{p_7} \cdot k_4 \Big|_{p_7} \cdot p_8,$$

$$X_{15} = x_{10} + \left| x_9 - x_{10} \right|_{p_9} \cdot k_5 \Big|_{p_9} \cdot p_{10},$$

$$X_{16} = x_{12} + \left| x_{11} - x_{12} \right|_{p_{11}} \cdot k_6 \Big|_{p_{11}} \cdot p_{12},$$

$$X_{17} = x_{14} + \left| x_{13} - x_{14} \right|_{p_{13}} \cdot k_7 \Big|_{p_{13}} \cdot p_{14},$$

$$X_{18} = x_{16} + \left| x_{15} - x_{16} \right|_{p_{15}} \cdot k_8 \Big|_{p_{15}} \cdot p_{16},$$

де $k_i = \left| p_{i+1}^{-1} \right|_{p_i}$; $x_1 - x_{16}$ – залишки по взаємно простих модулях $p_1 - p_{16}$,

– другий рівень:

$$X_{21} = X_2 + \left| X_1 - X_2 \right|_{p_{12}} \cdot k_{21} \Big|_{p_{12}} \cdot p_{34},$$

$$X_{22} = X_4 + \left| X_3 - X_4 \right|_{p_{56}} \cdot k_{22} \Big|_{p_{56}} \cdot p_{78},$$

$$X_{23} = X_6 + \left| X_5 - X_6 \right|_{p_{9_{-10}}} \cdot k_{23} \Big|_{p_{9_{-10}}} \cdot p_{11_{-12}},$$

$$X_{24} = X_8 + \left| X_7 - X_8 \right|_{p_{13_{-14}}} \cdot k_{24} \Big|_{p_{13_{-14}}} \cdot p_{15_{-16}},$$

де $k_{21} = \left| p_{34}^{-1} \right|_{p_{12}}$, $k_{22} = \left| p_{78}^{-1} \right|_{p_{56}}$, $k_{23} = \left| p_{11_{-12}}^{-1} \right|_{p_{9_{-10}}}$, $k_{24} = \left| p_{15_{-16}}^{-1} \right|_{p_{13_{-14}}}$,

$$P_{12} = p_1 \cdot p_2, P_{34} = p_3 \cdot p_4, P_{56} = p_5 \cdot p_6, P_{78} = p_7 \cdot p_8, P_{9_10} = p_9 \cdot p_{10},$$

$$P_{11_12} = p_{11} \cdot p_{12}, P_{13_14} = p_{13} \cdot p_{14}, P_{15_16} = p_{15} \cdot p_{16},$$

– третій рівень:

$$X_{31} = X_{22} + \left| X_{21} - X_{22} \right|_{P_{1_4}} \cdot k_{31} \Big|_{P_{1_4}} \cdot P_{5_8},$$

$$X_{32} = X_{24} + \left| X_{23} - X_{24} \right|_{P_{9_12}} \cdot k_{32} \Big|_{P_{9_12}} \cdot P_{13_16},$$

де $k_{31} = \left| P_{5_8}^{-1} \right|_{P_{1_4}}, \quad k_{32} = \left| P_{13_16}^{-1} \right|_{P_{9_12}}, \quad P_{1_4} = p_1 \cdot p_2 \cdot p_3 \cdot p_4,$

$$P_{5_8} = p_5 \cdot p_6 \cdot p_7 \cdot p_8,$$

– четвертий рівень:

$$X_{41} = X_{32} + \left| X_{31} - X_{32} \right|_{P_{1_8}} \cdot k_{41} \Big|_{P_{1_8}} \cdot P_{9_16},$$

де $k_{41} = \left| P_{9_16}^{-1} \right|_{P_{1_8}}, \quad P_{1_8} = p_1 \cdot p_2 \cdot p_3 \cdot p_4 \cdot p_5 \cdot p_6 \cdot p_7 \cdot p_8,$

$$P_{9_16} = p_9 \cdot p_{10} \cdot p_{11} \cdot p_{12} \cdot p_{13} \cdot p_{14} \cdot p_{15} \cdot p_{16}.$$

Структурна схема відновлення позиційного подання числа за n залишками приведена на рисунку 5.26.

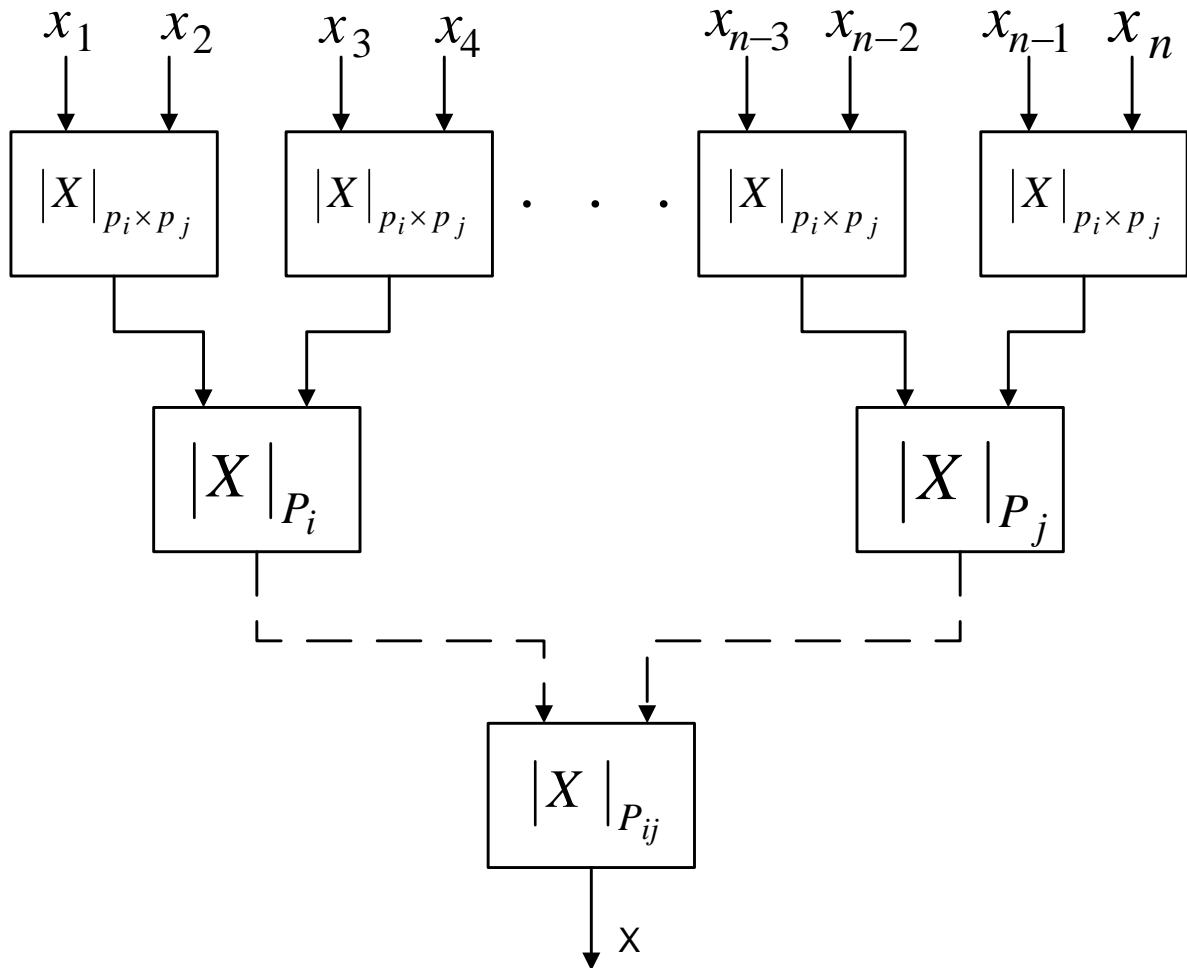


Рисунок 5.26 – Структурна схема відновлення позиційного подання числа за n залишками

Процес перетворення даних реалізовано на мові Verilog, моделювання та синтез пристрою проведено з використанням інструментального засобу Quartus II фірми Altera. Оскільки на входи $in0[12..0]-in15[12..0]$ ПЛІС поступають дані сенсорів, подані в СЗК, на виході $out[192..0]$ одержуємо багаторозрядний (максимальне значення становить 193 біти) двійковий код, який відображає дані 16 сенсорів (рисунок 5.27, 5.28).

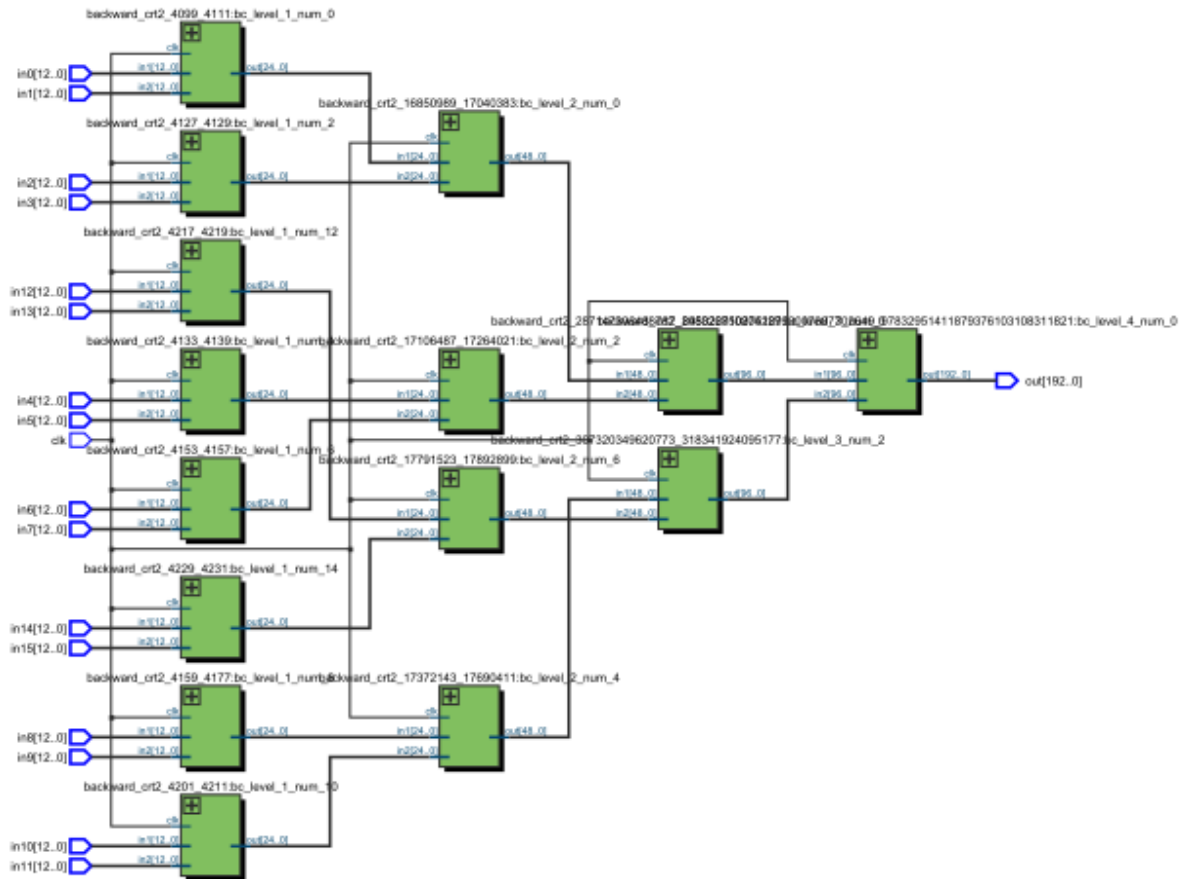


Рисунок 5.27 – Функціональна схема пристрою багаторівневого перетворення даних сенсорів

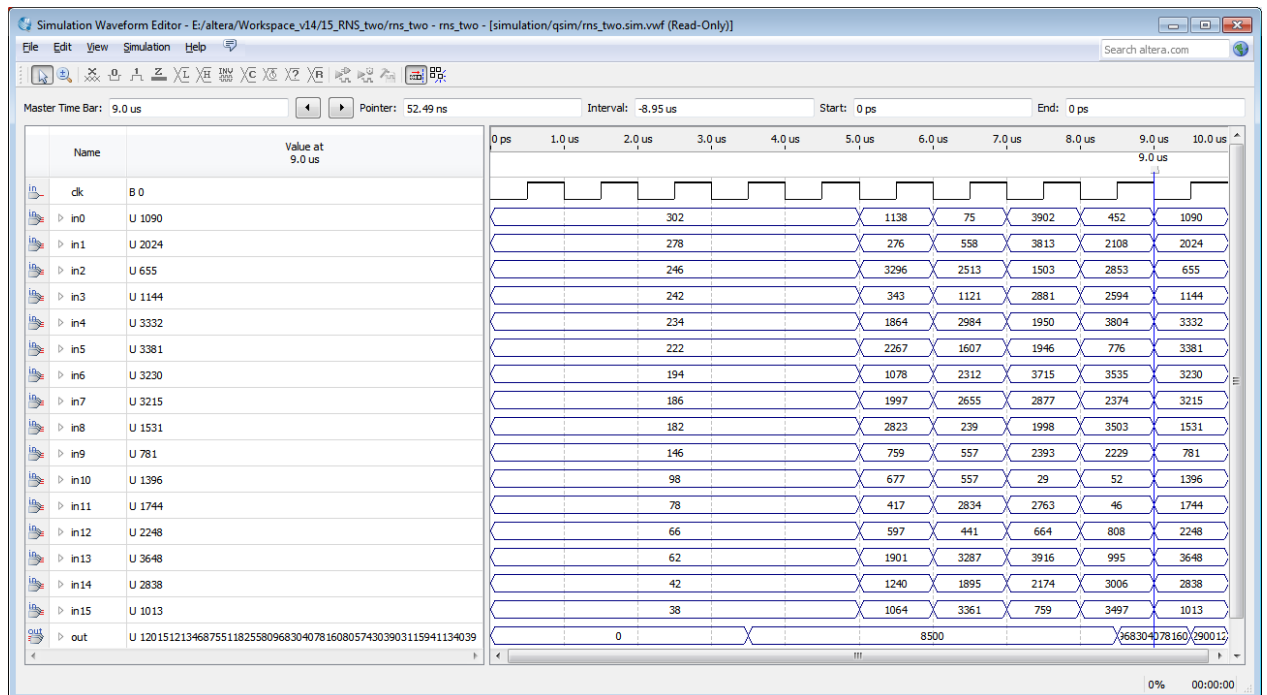


Рисунок 5.28 – Результати моделювання пристрою багаторівневого перетворення даних

Як видно з рисунку 5.28, значення на виході *out* відповідає системі залишків, які поступають на входи (*in0* – *in15*). Тобто, залишки (302, 278, 246, 242, 234, 222, 186, 182, 146, 98, 78, 66, 62, 42, 38) по модулях $p_1 - p_{16}$ дорівнюють 8500 в десятковій системі числення.

Апаратні затрати на реалізацію пристрою кодування з наявних в кристалі *EP4CGX110DF31C7* ресурсів становлять: логічні елементи – 34461 шт., регістри – 783 шт., вбудовані 9-ти бітні перемножувачі – 135 шт. (рисунок 5.29), максимальна затримка формування вихідних сигналів складає 17,264 нс (рисунок 5.30). Результати синтезу проекту в кристалі *EP4CGX110DF31C7* серії Cyclone IV GX фірми “Altera” подані на рисунку 5.29.

Flow Summary	
Flow Status	Successful - Wed Aug 12 19:40:5
Quartus II 64-Bit Version	14.0.0 Build 200 06/17/2014 SJ W
Revision Name	rns_two
Top-level Entity Name	reverse_conv_16
Family	Cyclone IV GX
Total logic elements	34,461 / 109,424 (31 %)
Total combinational functions	34,392 / 109,424 (31 %)
Dedicated logic registers	783 / 109,424 (< 1 %)
Total registers	783
Total pins	402 / 508 (79 %)
Total virtual pins	0
Total memory bits	0 / 5,621,760 (0 %)
Embedded Multiplier 9-bit elements	135 / 560 (24 %)
Total GXB Receiver Channel PCS	0 / 8 (0 %)
Total GXB Receiver Channel PMA	0 / 8 (0 %)
Total GXB Transmitter Channel PCS	0 / 8 (0 %)
Total GXB Transmitter Channel PMA	0 / 8 (0 %)
Total PLLs	0 / 8 (0 %)
Device	EP4CGX110DF31C7
Timing Models	Final

Рисунок 5.29 – Результати синтезу пристрою багаторівневого перетворення даних

Setup Times						
	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	▷ in15[*]	clk	17.264	17.772	Rise	clk
2	▷ in3[*]	clk	17.047	17.539	Rise	clk
3	▷ in7[*]	clk	16.948	17.489	Rise	clk
4	▷ in9[*]	clk	16.900	17.409	Rise	clk
5	▷ in14[*]	clk	16.612	17.156	Rise	clk
6	▷ in13[*]	clk	16.577	17.112	Rise	clk
7	▷ in8[*]	clk	16.426	17.135	Rise	clk
8	▷ in2[*]	clk	16.409	17.065	Rise	clk
9	▷ in11[*]	clk	16.350	16.994	Rise	clk
10	▷ in6[*]	clk	16.339	16.933	Rise	clk
11	▷ in10[*]	clk	15.900	16.506	Rise	clk
12	▷ in5[*]	clk	15.879	16.469	Rise	clk
13	▷ in12[*]	clk	15.675	16.291	Rise	clk
14	▷ in4[*]	clk	15.317	15.957	Rise	clk
15	▷ in1[*]	clk	14.761	15.305	Rise	clk
16	▷ in0[*]	clk	13.936	14.580	Rise	clk

Рисунок 5.30 – Часові характеристики роботи пристрою перетворення

Наведений приклад проектування пристрою перетворення даних сенсорів, поданих в СЗК, може бути легко адаптований для іншої розрядності даних сенсорів та їх кількості.

Таким чином, апаратні затрати на реалізацію пристрою перетворення в кристалі *EP4CGX110DF31C7*, серії Cyclone IV GX становлять: логічні елементи – 34461 шт., регістри – 783 шт., вбудовані 9-ти бітні перемножувачі – 135 шт., максимальна затримка формування вихідних сигналів складає 17,264 нс.

5.5 Програмний модуль генератора Verilog-коду

Для прискорення розробки декодера модулярних коректуючих кодів та виключення помилок при написанні Verilog-коду модуля виправлення помилок розроблено програмне забезпечення для автоматичного генерування Verilog-коду по заданих параметрах.

При розробці програмного коду на мові JavaScript була використана JavaScript-бібліотека JQuery з відкритим сирцевим кодом. Синтаксис jQuery забезпечує зручне орієнтування у навігації, завдяки вибору елементів DOM, створення анімації, обробки подій, і розробки AJAX- додатків.

Весь функціонал генератора розроблений та підключений у файлі script.js:

```
<script type="text/javascript" src="script.js"></script>
```

Інтерфейс користувача складається із чотирьох полів та відповідних підказок до них:

```
<form class="parform" name="parform">
  <div class="form-group">
    <input type="text" id="quantity" name="quantity" value=""
      placeholder="Кількість символів">
  </div>
  <div class="form-group">
    <input type="text" id="bit" name="bit" value=""
      placeholder="Розрядність символів">
  </div>
  <div class="form-group">
    <input type="text" id="coefficient" name="coefficient"
      value="" placeholder="Коефіцієнти (наприклад: 3,5,7...)">
  </div>
  <div class="form-group">
    <input type="text" id="module" name="module" value=""
      placeholder="Модуль">
  </div>
  <div class="form-group">
    <input class="btn" type="button" id="submit" value="Generate
code">
    <input class="btn" type="reset" id="reset" value="Reset">
  </div>
</form>
```

Генерування Verilog-коду починається після натискання на кнопку «Submit», при цьому відбувається запуск JavaScript коду:

```
$('#submit').click(function(){...code...})
```

Для зчитування введених даних використовується JQuery метод «.val()»

```
var user_quantity = $('#quantity').val();
var bit_in = $('#bit').val();
var module = $('#module').val();
var coefficient = $('#coefficient').val();
```

При введенні вхідних параметрів здійснюється перевірка їх коректності, зокрема виконується перевірка довжини отриманої стрічки із даними.

```
if (user_quantity.length<=0 || bit_in.length<=0 || module.length<=0 ||
coefficient.length<=0){
    alert("Заповніть всі поля");
    return;
}
```

Для генерації коду основного модуля створені масиви в які за допомогою методу «.push()» записуються дані, згенеровані через цикл for. Для запису даних у блок виводу використано JQuery метод «.append()»:

```
var quantity = [];
var quantity_mod = [];
var quantity_res = [];
var sindrom = [];
var bit = bit_in-1;
coefficient = coefficient.split(',');
for (var i = 1; i<=user_quantity; i++) {
    quantity.push("b0" + i);
    quantity_mod.push("b" + i);
    quantity_res.push("\tno_error_" + i + "("+quantity_mod[i-1]+", "+quantity[i-1]+",s);\n")
}
```

```

$('.output-con .container').append('<div class="output">'+
    '<pre>module korec_3m ('+quantity+', '+quantity_mod+',s);\n'+
    '\tinput ['+bit+':0] '+quantity+';\n'+
    '\tinput wire [5:0] s;\n'+
    '\toutput wire ['+bit+':0]'+quantity_mod+';\n'+
    '<br>'+
    quantity_res.join(" ")
    '<br>endmodule<br></pre>'+
    '</div>');

```

Розрядність вхідних символів обмежена діапазоном від 3 до 8 біт, який реалізовано за допомогою циклу «switch»:

```

var bit_length;
switch (bit_in){
    case '3':
        bit_length = 7;
        break;
    case '4':
        bit_length = 15;
        break;
    case '5':
        bit_length = 31;
        break;
    case '6':
        bit_length = 63;
        break;
    case '7':
        bit_length = 127;
        break;
    case '8':
        bit_length = 255;
        break;
}

```

Блоки коду для виявлення та виправлення помилок генеруються аналогічно до основного модуля:



Генератор Verilog коду

Кількість символів

Розрядність символів

Коефіцієнти (наприклад: 3,5,7...)

Модуль

Generate code

Reset

Рисунок 5.31 – Інтерфейс генератора Verilog-коду

Для тестування обрано наступні вхідні параметри декодера для генерації коду: кількість символів – 8; розрядність символів – 4 біти; коефіцієнти – 13, 17, 19, 23, 29, 31, 37, 43; модуль – 1021 (рисунок 5.32).



Генератор Verilog коду

8

4

13, 17, 19, 23, 29, 31, 37, 43

1021

Generate code

Reset

Рисунок 5.32 – Інтерфейс генератора із введеними параметрами коду

В результаті виконання JavaScript коду генерується Verilog-код, який виводиться у блок output. Приклад Verilog-коду для виправлення помилок в першому блоці даних наведений на рисунку 5.33. Verilog-код основного модуля наведено на рисунку 5.34.

```
module no_error_1(b1,b01,s);
  output reg[3:0] b1;
  input [3:0] b01;
  input [11:0] s;

  always @(*)
  begin
    if (s==195)
    begin
      b1=(b01 - 15);
    end
    else
    if (s==182)
    begin
      b1=(b01 - 14);
    end
    else
    if (s==169)
    begin
      b1=(b01 - 13);
    end
    .
    .
    .
    else
    if (s==826)
    begin
      b1=(b01 + 15);
    end
    else
    begin
      b1=b01 [3:0];
    end
  end
endmodule
```

Рисунок 5.33 – Код виправлення помилок в першому блоці даних

```
module korec_3m (b01,b02,b03,b04,b05,b06,b07,b08,b1,b2,b3,b4,b5,b6,b7,b8,s);  
  input [3:0] b01,b02,b03,b04,b05,b06,b07,b08;  
  input wire [11:0] s;  
  output wire [3:0] b1,b2,b3,b4,b5,b6,b7,b8;  
  
  no_error_1(b1,b01,s);  
  no_error_2(b2,b02,s);  
  no_error_3(b3,b03,s);  
  no_error_4(b4,b04,s);  
  no_error_5(b5,b05,s);  
  no_error_6(b6,b06,s);  
  no_error_7(b7,b07,s);  
  no_error_8(b8,b08,s);  
  
endmodule
```

Рисунок 5.34 – Verilog-код основного модуля

Аналогічно формуються коди для виправлення помилок в інших блоках даних, кількість модулів коду дорівнює кількості символів в блоці даних.

Проведене тестування показало правильну роботу генератора Verilog-коду. Повний код генератора наведений в додатку Е. Використання розробленого генератора дозволяє миттєво і без помилок згенерувати код модуля виправлення помилок із заданими параметрами.

Висновки до п'ятого розділу

1. Проведено аналіз та експериментальні дослідження швидкодії виконання та апаратних затрат при реалізації методів виконання модулярних операцій на ПЛІС. Проведені дослідження показали, що використання методу різниці квадратів підвищує швидкодію виконання операції множення в середньому на 40%. Експериментально доведено, що при розрядності модулів (3-5 біт) апаратні затрати в 2 - 3 рази менші, порівняно із апаратними

затратами при розрядності модулів 8-10 біт при однаковому діапазоні подання чисел.

2. Розроблено структури та реалізовано на ПЛІС кодер/декодер, який виявляє і виправляє однократні помилки на основі модулярного коректуючого коду. Робота кодера і декодера описана на мові Verilog в середовищі Quartus II та реалізована на ПЛІС фірми “Altera”, серії MAX II (кодер) та Cyclone IV GX (декодер). Апаратні затрати при реалізації на ПЛІС фірми Altera, серії MAX II становлять: кодера – 58 логічних елементів, максимальна затримка формування коректуючого символу становить 42,74 нс.; декодера – 1108 логічних елементів, максимальна затримка формування вихідних сигналів становить 62,6 нс.

3. Розроблено структуру спецпроцесора виявлення та виправлення двократних помилок на основі модулярних коректуючих кодів. Розроблено алгоритми виявлення, локалізації та виправлення помилок. Для підвищення швидкодії виправлення помилок пошук мультиплікативних обернених елементів реалізовано у вигляді таблиць. Проведена оцінка апаратних затрат при використанні різних серій ПЛІС фірми “Altera” показала, що для реалізації декодера на мікросхемі EP4CGX22CF19C6, серії Cyclone IV GX без використання вбудованих блоків необхідно 6986 логічних елементів.

4. Розроблено пристрій багаторівневого перетворення даних в системі залишкових класів, який забезпечує об’єднання 12 – ти розрядних даних від 16 сенсорів. Перетворення вхідних даних сенсорів, поданих в СЗК, в двійкову систему числення реалізовано на основі попарного об’єднання залишків по двох модулях, що дозволило зменшити розрядність даних в процесі виконання проміжних обчислень, що в свою чергу зменшило апаратні затрати. Апаратні затрати на реалізацію пристрою перетворення із наявних ресурсів в кристалі *EP4CGX110DF31C7*, серії Cyclone IV GX становлять: логічні елементи – 34461 шт., регістри – 783 шт., вбудовані 9-ти

бітні перемножувачі – 135 шт., максимальна затримка формування вихідних сигналів складає 17,264 нс.

5. Створено генератор Verilog-коду для синтезу на ПЛІС декодера модулярних коректуючих кодів, який автоматично генерує Verilog-код по заданих параметрах, що дозволило підвищити швидкодію розробки та виключити помилки при написанні коду.

РОЗДІЛ 6

РОЗРОБКА, МОДЕЛЮВАННЯ ТА ВПРОВАДЖЕННЯ УДОСКОНАЛЕНИХ БЕЗПРОВІДНИХ СЕНСОРНИХ МЕРЕЖ

6.1 Моделювання роботи мережі при використанні схем контролю помилок

Для підвищення надійності передачі даних в БСМ з багатоінтервальною маршрутизацією використовують два підходи до виправлення помилок: 1) контроль помилок в кінцевому вузлі; 2) контроль помилок в проміжних вузлах маршруту [123, 208].

З метою виявлення ефективних схем застосування, розроблених в розділі 3 модулярних коректуючих кодів, проведена оцінка затрат енергії при використанні вказаних вище підходів виявлення помилок.

1) Контроль помилок в кінцевому вузлі. При даному методі передачі дані кодуються у вихідному вузлі з використанням коректуючих кодів і декодуються в кінцевому вузлі (базовій станції). Проміжні вузли тільки ретранслюють пакети даних. Однак, при такій організації передачі відбувається накопичення помилок в пакеті, що унеможлиблює декодування даних у кінцевому вузлі, що призводить до необхідності повторної передачі пакетів даних. Даний метод зменшує затрати енергії при відсутності помилок в каналі зв'язку, однак потребує додаткових досліджень затрат енергії при високій ймовірності помилок.

Для дослідження коефіцієнту бітових помилок (BER) в БСМ, фізичний рівень якої відповідає стандарту IEEE 802.15.4, розроблена модель каналу передачі даних мережі ZigBee. Модель складається з генератора даних (генератора випадкових чисел), ZigBee передавача, каналу зв'язку, ZigBee приймача, блока обчислення помилок та індикатора (рисунок 6.1).

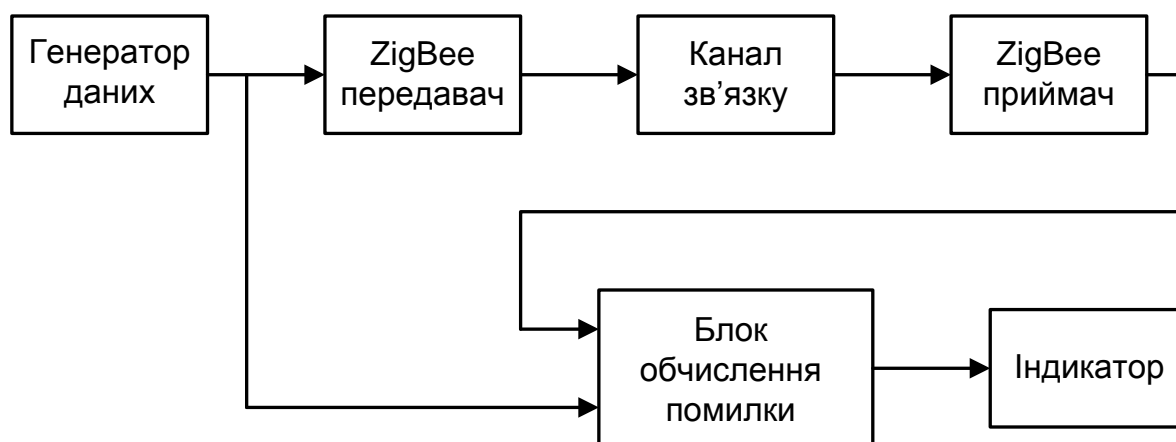


Рисунок 6.1 – Модель каналу передачі даних мережі ZigBee

Розрахунок затрат енергії на передачу даних в безпроводних мережах проведено на основі аналітичної моделі, наведеної в роботі [165].

Ймовірність правильного отримання пакету даних при багатоінтервальної маршрутизації визначається [208]

$$P = (1 - BER)^H,$$

де BER – ймовірність помилки в каналі;

H – кількість проміжних вузлів у маршруті.

Відповідно, кількість пакетів, отриманих з помилкою:

$$n_{err} = (1 - P) \cdot n_{pac}, \quad (6.1)$$

де n_{pac} – загальна кількість переданих пакетів.

Загальна затрата енергії на передачу n_{pac} пакетів визначається:

$$E_t = \sum_{i=1}^H E_b \cdot (n_{pac} + n_{err}), \quad (6.2)$$

де E_b – затрати енергії для передачі одного біту.

Затрати енергії для передачі одного біту [198]

$$E_b = \frac{(I_t + I_r) \cdot U \cdot n}{C_b},$$

де I_t – струм споживання передавача;

I_r – струм споживання приймача;

U – напруга живлення передавача і приймача;

n – кількість бітів в пакеті;

C_b – швидкість передачі даних в мережі.

Параметри безпроводних модулів та мережі, використані для моделювання затрат енергії на передачу даних, наведені в таблиці 6.1.

Таблиця 6.1 – Параметри безпроводних модулів та мережі

Параметр	I_t , мА	I_r , мА	U , В	n	n_{pac}	C_b , кбіт/с
Значення	15	17	3,0	1016	100	250

Теоретичні значення залежності ймовірності помилки від значення E_b/N_0 для БСМ на основі стандарту IEEE 802.15.4 отримані з використанням Communications System Toolbox системи Matlab і наведені на рисунку 6.2.

На основі формули (6.1) розраховано залежність кількості пакетів, прийнятих з помилками, від відношення E_b/N_0 для кількості проміжних вузлів у маршруті $H = 2, \dots, 10$ (рисунок 6.3).

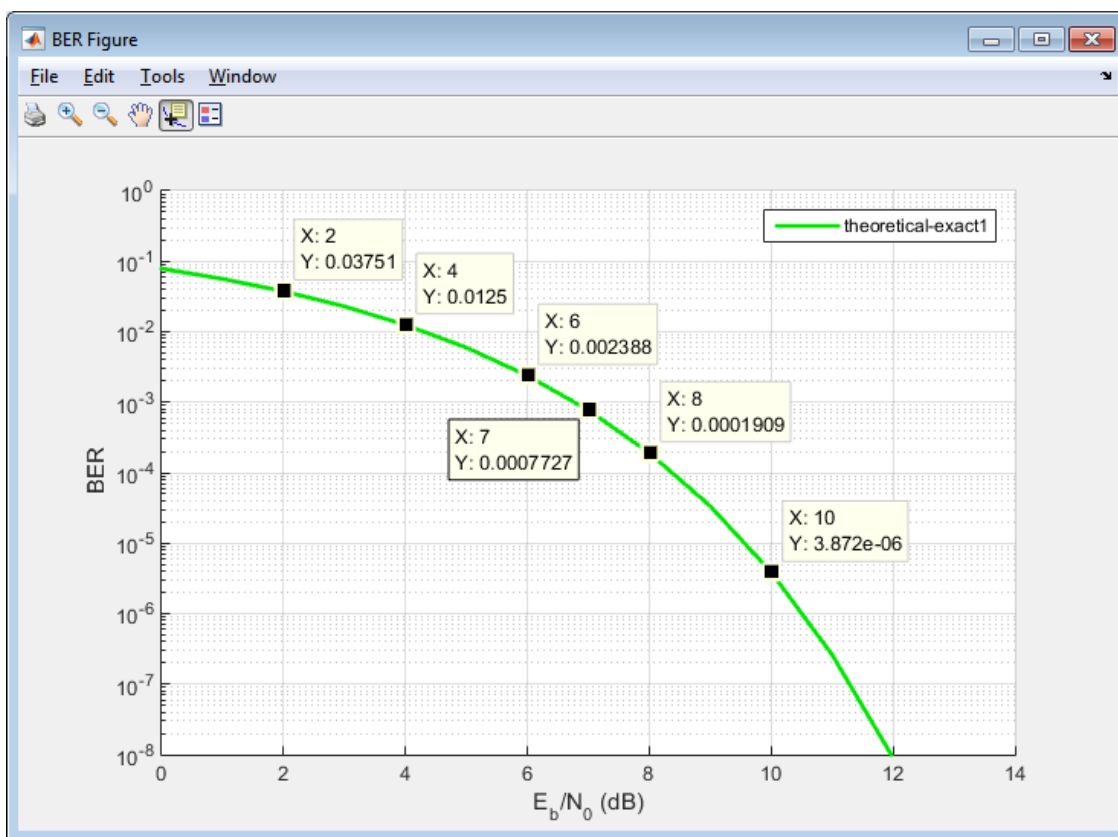


Рисунок 6.2 – Залежність ймовірності помилки від значення E_b/N_0 : E_b – енергія біту, N_0 – спектральна густина потужності шуму.

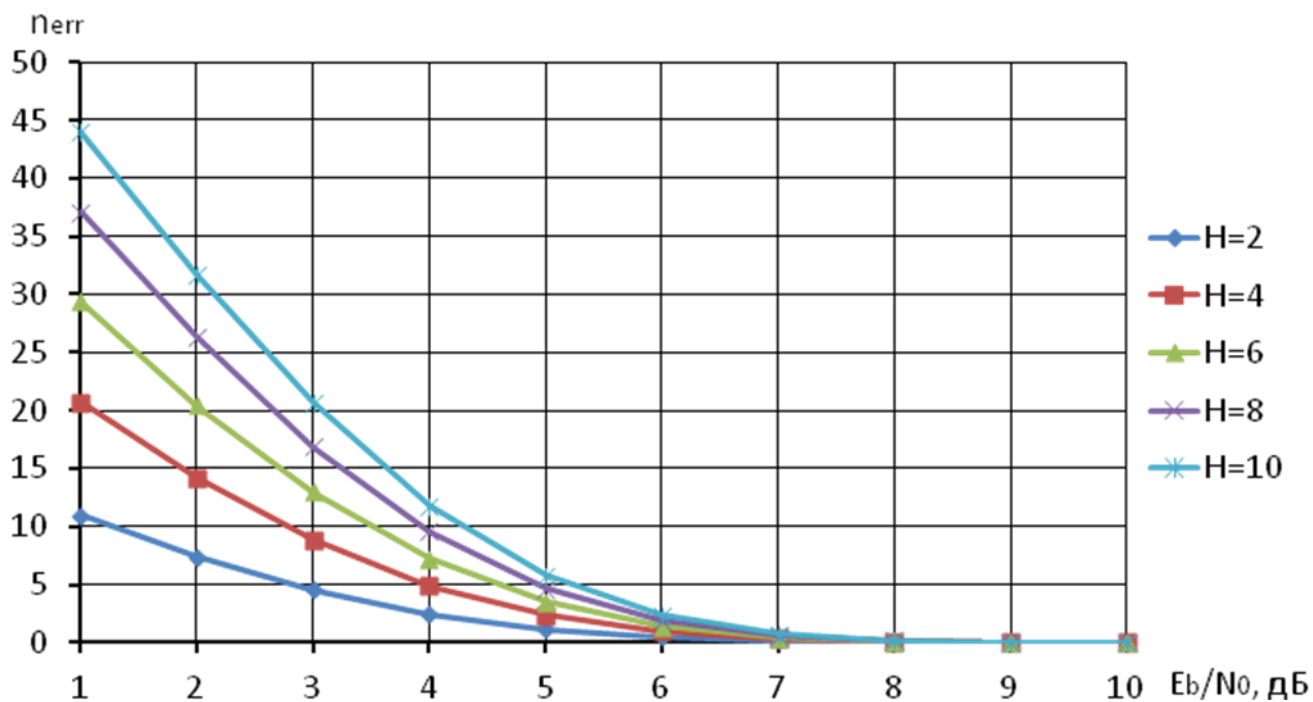


Рисунок 6.3 – Кількість отриманих пакетів даних з помилками в залежності від значення E_b/N_0 для маршрутів з різною кількістю вузлів

Як видно з рисунку 6.3, при $E_b/N_0 < 7$, що відповідає ймовірності помилки $BER \approx 10^{-3}$ (див. рисунок 6.2) різко зростає кількість пакетів, прийнятих з помилкою.

На основі формули (6.2) та таблиці 6.1 розраховано затрати енергії на передачу пакету даних в залежності від значення E_b/N_0 при різній кількості вузлів в маршруті (рисунок 6.3). Як видно з рисунку 6.4, затрати енергії на повторну передачу пакетів, отриманих з помилками, в основному залежать від кількості проміжних вузлів і зростають на 10-40 % в залежності від значення E_b/N_0 .

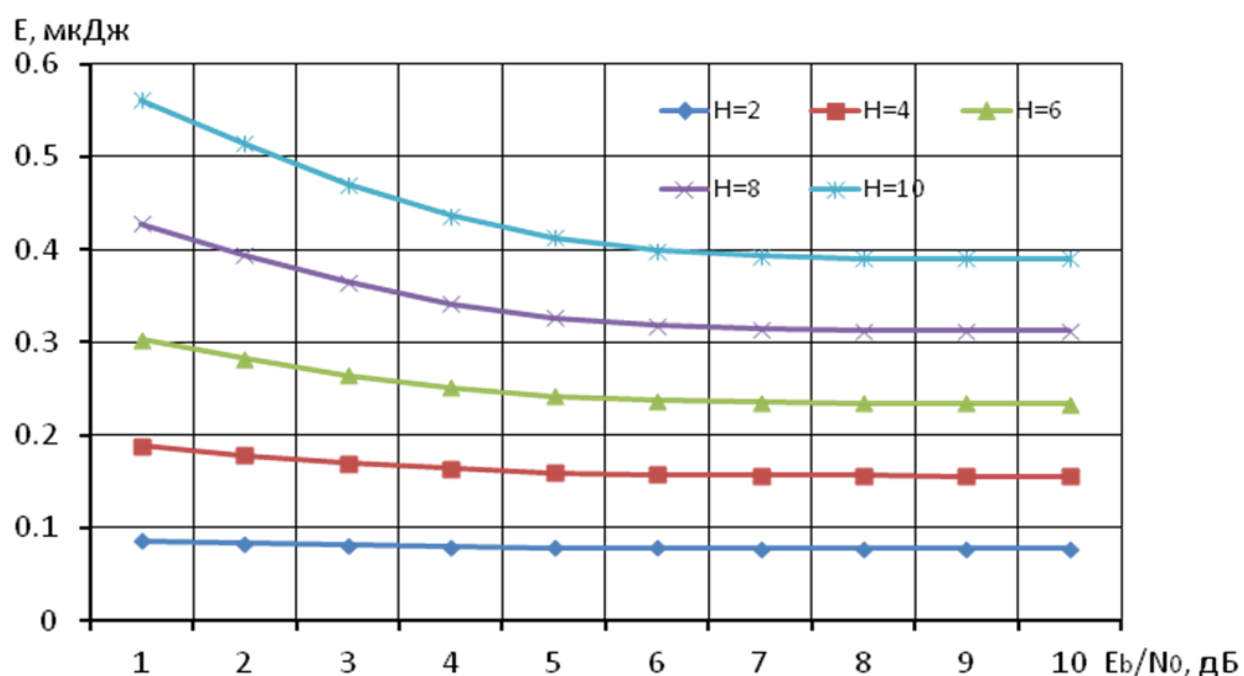


Рисунок 6.4 – Затрати енергії на передачу пакету даних в залежності від значення E_b/N_0 при різній кількості вузлів в маршруті

Другий метод контролю помилок передбачає декодування пакетів в усіх проміжних вузлах маршруту.

В даному випадку ймовірність отримання пакету даних з помилкою:

$$P_{er} = (1 - BER), \quad (6.3)$$

відповідно, кількість пакетів, отриманих з помилкою:

$$n_{err} = (1 - P_{er}) \cdot n_{pac}. \quad (6.4)$$

Енергія на передачу n_{pac} пакетів даних між двома вузлами мережі:

$$E_{ntn} = n_{pac} \cdot r \cdot (E_b + E_{dec}) + n_{err} \cdot r \cdot (E_b + E_{dec}), \quad (6.5)$$

де r – коефіцієнт, який враховує надлишковість коректуючого коду;

$E_{dec} = 0.3 \cdot E_b$ – затрати енергії на декодування даних.

Загальна енергія на передачу n_{pac} пакетів даних через H проміжних вузлів у мережі:

$$E_{total} = \sum_{i=1}^H E_{ntn}.$$

На основі формул (6.3 – 6.5) побудовано залежності затрат енергії на передачу пакету даних від значення E_b / N_0 та швидкості коду (рисунок 6.5).

Як видно з рисунку 6.5, при значенні $E_b / N_0 < 3$ використання коректуючих кодів з $R = 0,4$ забезпечує менші затрати енергії.

Отже, використання коректуючих кодів з декодуванням в проміжних вузлах має переваги при рівні завад ($BER > 10^{-3}$) і вище, а метод ретрансляції пакетів (з декодуванням в кінцевому вузлі) має переваги при рівні завад ($BER < 10^{-3}$) та передачі даних, чутливих до затримок.

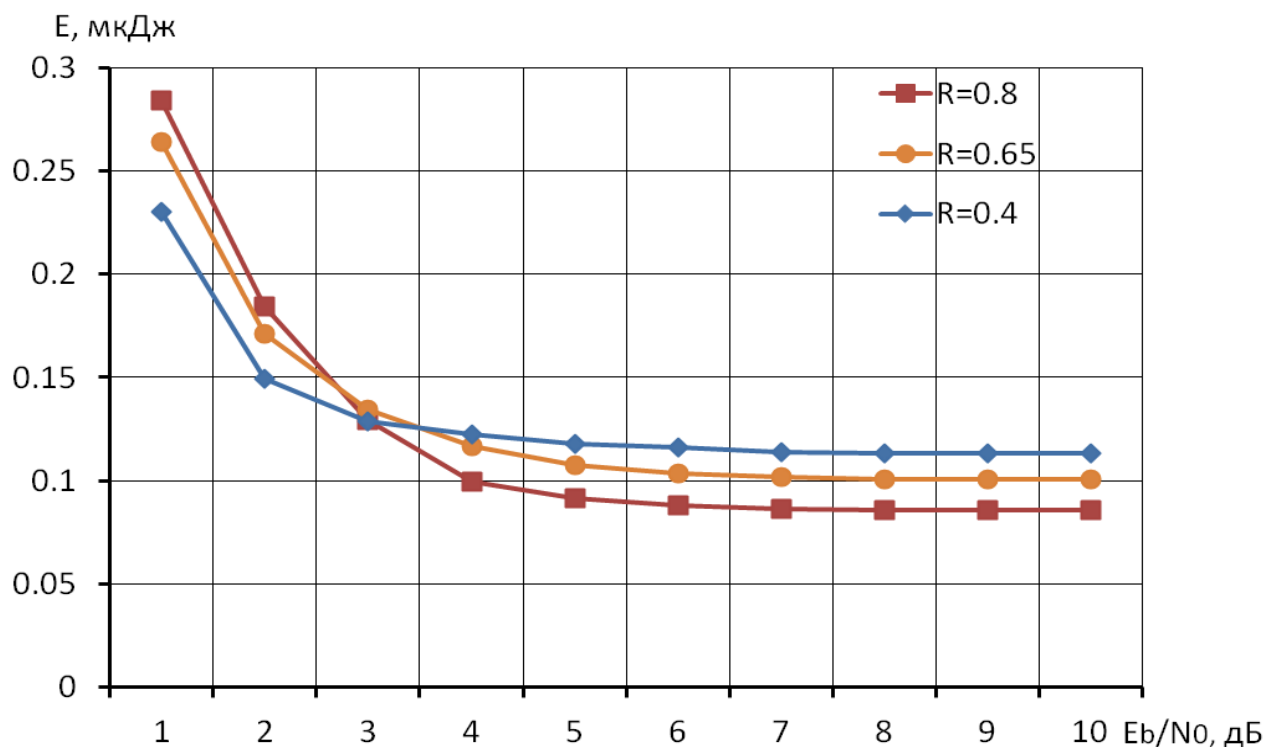


Рисунок 6.5 – Затрати енергії на передачу пакету даних в залежності від значення E_b/N_0 для швидкості коду: 0,8; 0,65; 0,4.

Отримані результати моделювання враховують тільки теоретичне значення BER, тому для визначення реального стану каналу зв'язку необхідно провести експериментальні дослідження з врахуванням роботи інших радіопередавачів в діапазоні частот 2,4 ГГц.

6.2 Експериментальне тестування каналу зв'язку безпроводної сенсорної мережі

Для вибору ефективної схеми контролю помилок в БСМ розроблено методику та програму тестування, яка визначає кількість втрачених пакетів Packet Error Rate (PER) в мережі ZigBee.

Суть тестування полягає в тому, що пакети надсилаються від одного ZegBee пристрою до іншого, при цьому кількість втрачених пакетів підраховується і відображається на індикаторі або зберігається в файлі при з'єднанні з ноутбуком (рисунок 6.6).

Значення PER визначається, як число пакетів, прийнятих з помилкою, N_{error} до загального числа прийнятих пакетів N_0 :

$$PER = \frac{N_{error}}{N_0} \cdot 100, \%$$

Пакет вважається прийнятим з помилкою, якщо хоча б один біт є спотвореним.

Програма підтримує два режими роботи: без відправки пакету підтвердження (рисунок 6.6) і з відправкою пакету підтвердження (рисунок 6.7).

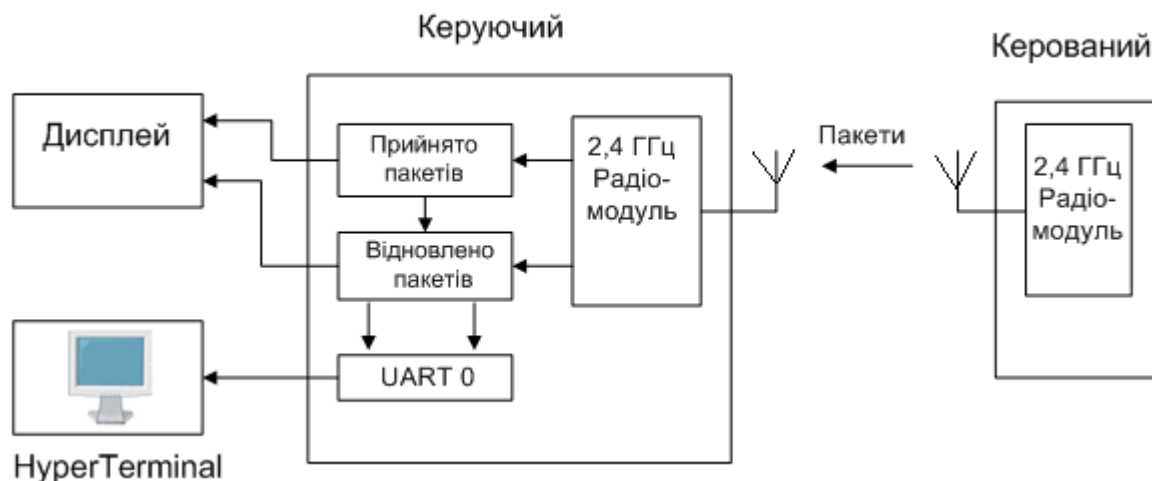


Рисунок 6.6 – Схема тестування каналу зв'язку БСМ без відправки пакету підтвердження

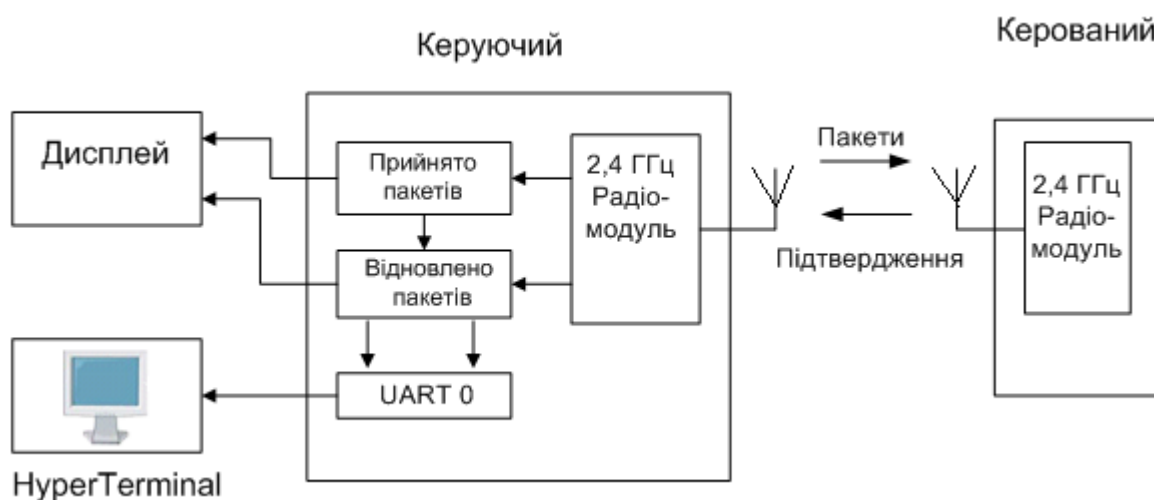


Рисунок 6.7 – Схема тестування каналу зв'язку БСМ з відправкою пакету підтвердження

Експерименти проводились на відлагоджувальному комплекті фірми Jennic JN5148-EK010 з використанням безпроводних мікроконтролерів JN5148 001-M00 з вбудованою керамічною антеною, JN5148-001-M03 з роз'ємом для підключення антени і JN5148-001-M04 з антенним роз'ємом та підсилювачем потужності для збільшення дальності передачі.

За допомогою кнопок SW1 – SW4 на платі Master та відповідного пункту меню програма дозволяє вибирати наступні налаштування.

1. Меню «Channel». Вибір каналу зв'язку. Номер каналу на обох платах можна збільшити шляхом натискання кнопки SW1 на платі Master, якщо номер каналу приймає значення 25, то наступне натискання кнопки переводить його до каналу з номером 11.

2. Рівень потужності. Використовуються чотири допустимі значення рівня потужності для модулів JN5148 – це 0, 8, 16 і 24, які відповідають PAP = 0, 1, 2 і 3 (таблиця 6.2).

Таблиця 6.2 – Рівень потужності модулів

Рівень потужності	0	8	16	24
Стандартний модуль	-32 дБм	-21 дБм	-9,5 дБм	2 дБм
Модуль високої потужності	-13 дБм	-1 дБм	9 дБм	16 дБм

3. Меню «Retries» використовується для встановлення кількості спроб повторної передачі пакету. Кожне натискання кнопки SW3 збільшує значення кількості спроб на одиницю (максимум 7, наступне натискання приведе до повернення в нуль).

4. Меню «Mode» встановлює режим роботи без передачі пакету підтвердження або з передачею пакету підтвердження.

5. Для відображення потужності сигналу на платі «Майстер», використовуються чотири світлодіоди D1, D2, D3 і D4:

– 4 світлодіоди світяться – потужність прийнятого сигналу більше 60 дБ вище рівня чутливості приймача;

– 3 світлодіода світяться – потужність прийнятого сигналу більше 40 дБ вище рівня чутливості приймача;

– 2 світлодіоди світяться – потужність прийнятого сигнал більше 20 дБ вище рівня чутливості приймача;

– 1 світлодіод світиться – потужність прийнятого сигналу менше 20 дБ вище рівня чутливості приймача;

– 0 світлодіодів світиться – сигнал відсутній.

Значення потужності сигналу відображається на індикаторі у вигляді графіка (рисунок 6.8) LQI (Link Quality Indication Quality) та виводиться в послідовний порт у вигляді цифрового значення.

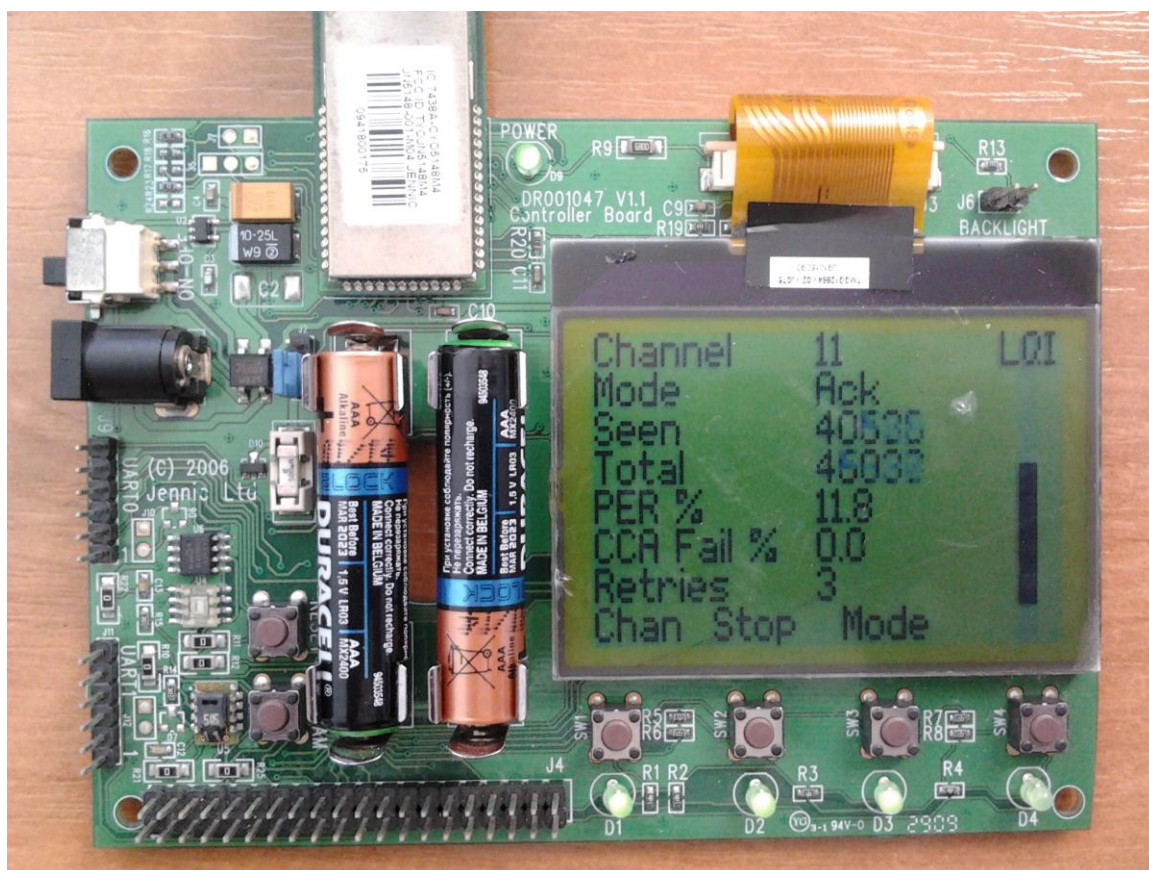


Рисунок 6.8 – Модуль тестування з цифровим індикатором

Для відображення результатів на ПК використовується NurerTerminal з налаштуваннями: 38400 біт/с, 8 біт, без контролю парності, 1 стоп-біт, без управління потоком.

На дисплеї відображається поточний канал (Channel – 11), кількість успішно прийнятих пакетів (Seen), загальна кількість відправлених кадрів (Total) і відсоток пакетів, отриманих з помилкою PER (де 0% означає всі пакети прийняті правильно і 100% всі пакети прийняті з помилкою). CCA fail показує відсоток пакетів, які не вдалося передати через те, що канал зашумлений роботою інших пристроїв (Wi-fi, Bluetooth).

Проведено дослідження впливу роботи Wi-fi мережі на кількість пакетів, прийнятих з помилкою в мережі ZigBee (рисунок 6.9), яке показало, що кількість пакетів, прийнятих з помилкою, становить від 3,4 – 10,6% для різних частотних каналів. Найбільший відсоток пакетів, прийнятих з

помилкою, (10,6; 9,0) на 22 і 23 каналах, що відповідає каналу № 10 мережі Wi-fi (рисунок 6.10).

PER, %

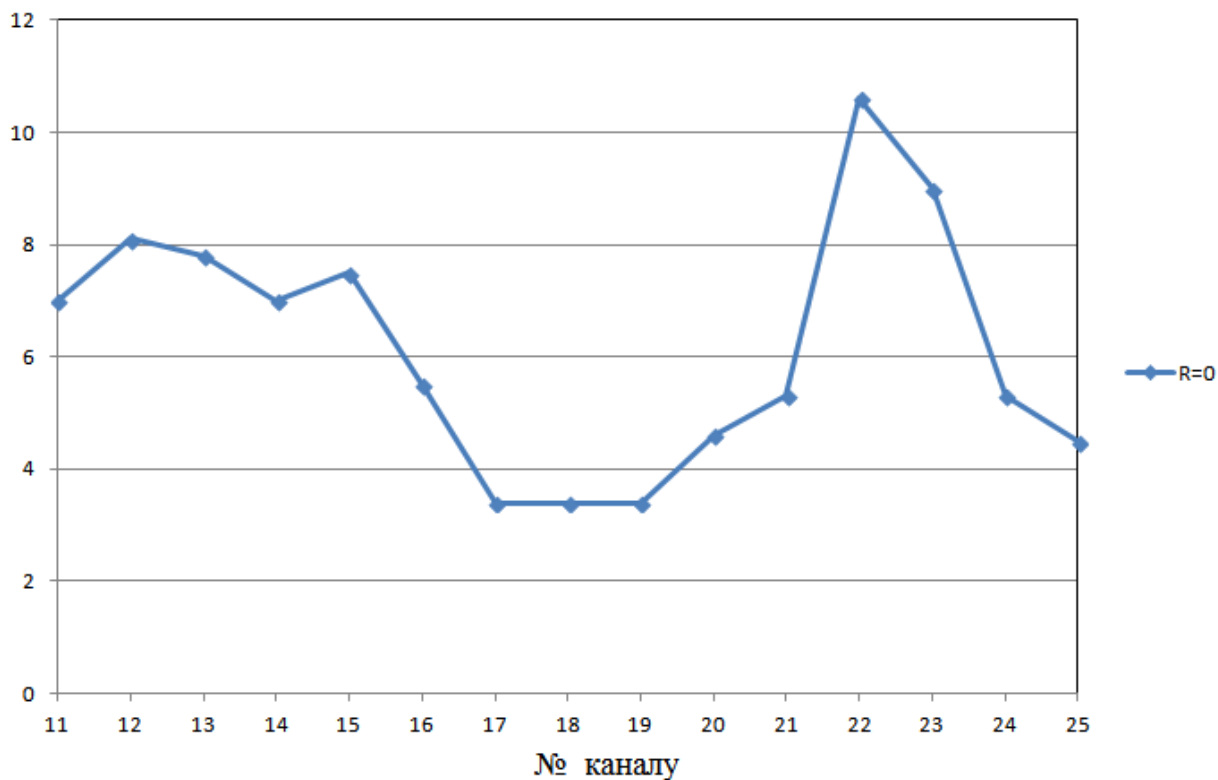


Рисунок 6.9 – Кількість прийнятих з помилкою пакетів на різних частотних каналах

Wi-fi аналізатор

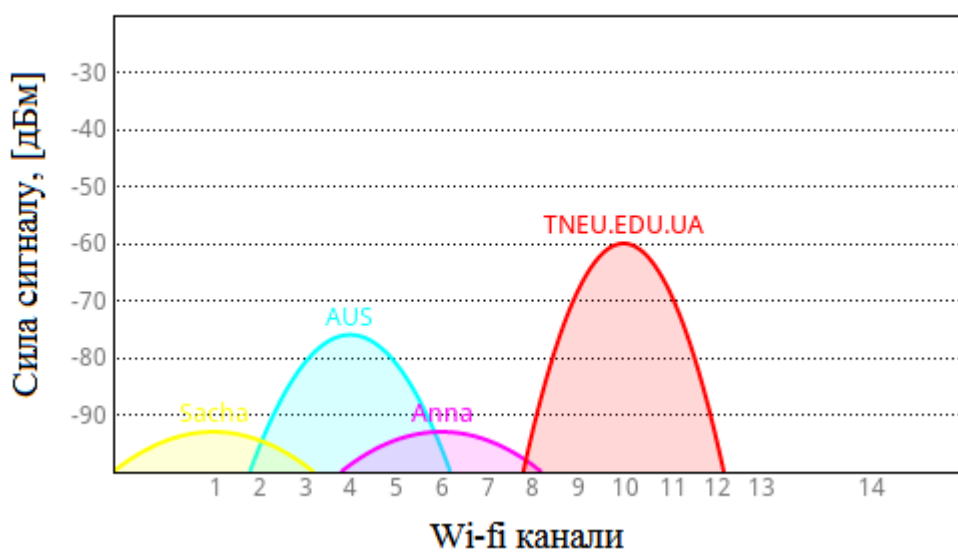


Рисунок 6.10 – Потужність сигналу Wi-fi мережі на території проведення експерименту

Використання повторних запитів на передачу пакетів, отриманих з помилкою, дозволяє зменшити відсоток прийнятих з помилкою пакетів в середньому в 20 разів при $R=3$ (три повторних запити) і ще в 2,5 рази, порівняно з попереднім значенням, при $R=7$ (рисунок 6.11).

Результати тестування каналу зв'язку БСМ показали, що при виборі коректуючих кодів та схем контролю помилок необхідно враховувати вплив обладнання, яке працює в діапазоні частот 2,4 ГГц (Wi-fi, Bluetooth та інше).

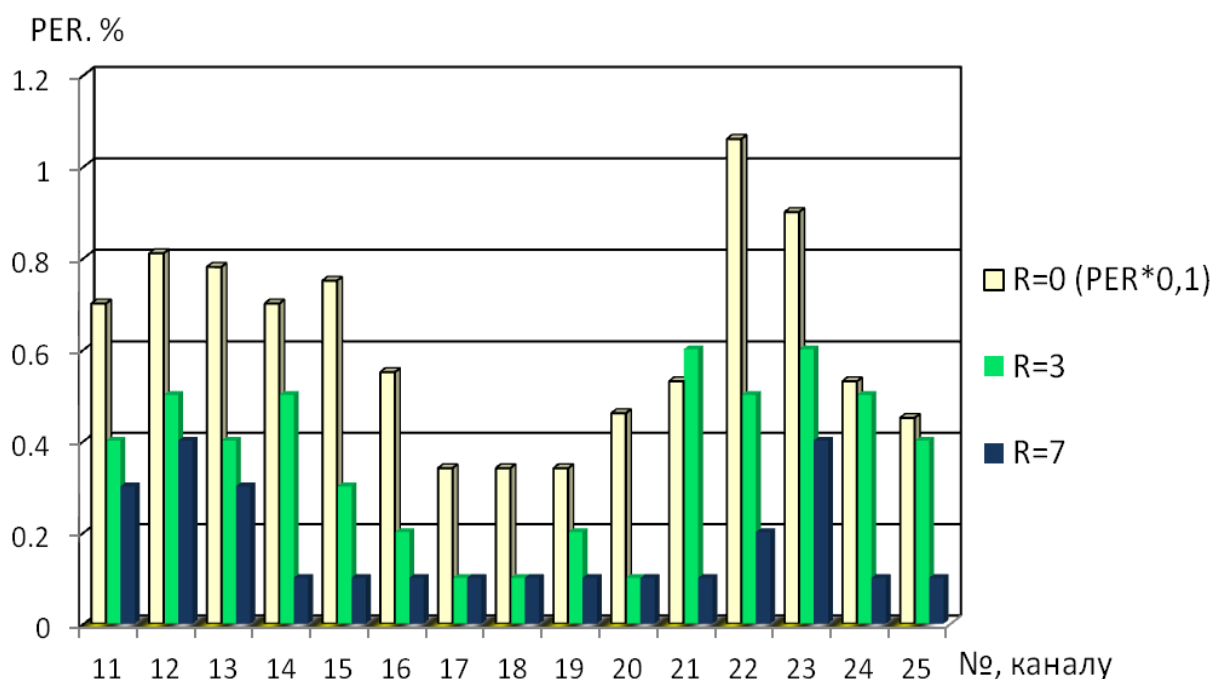


Рисунок 6.11 – Кількість пакетів, прийнятих з помилкою, при використанні різної кількості повторних запитів (для $R=0$ наведені значення взято з коефіцієнтом 0,1)

Отримані експериментальні результати тестування каналу зв'язку БСМ дозволяють вибрати необхідні параметри розроблених модулярних коректуючих кодів з в залежності від стану каналу передачі.

Дану методику тестування доцільно використовувати для оцінки надійності передачі даних в БСМ, які працюють в різних умовах та середовищах передачі даних (високий рівень електромагнітних завад) на

етапі розробки (для вибору коректуючих кодів) та вводу в експлуатацію БСМ.

6.3 Безпроводна сенсорна мережа системи відеоспостереження об'єктів

Безпроводні мультимедійні сенсорні (БМСМ) мережі мають значний потенціал, щоб стати основою для створення багатьох нових додатків. Однак, при розробці архітектури БМСМ необхідно враховувати ряд факторів, зокрема [77, 99, 113, 175]:

- вимоги до якості обслуговування QoS;
- обмеженість ресурсів;
- змінна пропускна здатність каналів;
- крос-рівнева взаємодія функцій.

Мультимедійні дані чутливі до втрати пакетів, зокрема, втрата невеликої частини даних зображення призводить до відкидання всього зображення або до різкого зниження його якості.

Враховуючи обмежений розмір поля даних структури протоколів БМСМ, мультимедійний контент необхідно розділяти на велику кількість пакетів, які не повинні бути втрачені або спотворені в процесі їх передачі для відновлення зображення.

Розроблена БМСМ складається з безпроводних вузлів, обладнаних CMOS камерою та базовою станцією (БС) (рисунок 6.12). Базова станція, як правило, забезпечена постійним живленням та високошвидкісним підключенням до мережі Інтернет. В залежності від умов застосування БМСМ роль базової станції виконує звичайний або спеціалізований комп'ютер, обладнаний безпроводним радіомодулем.

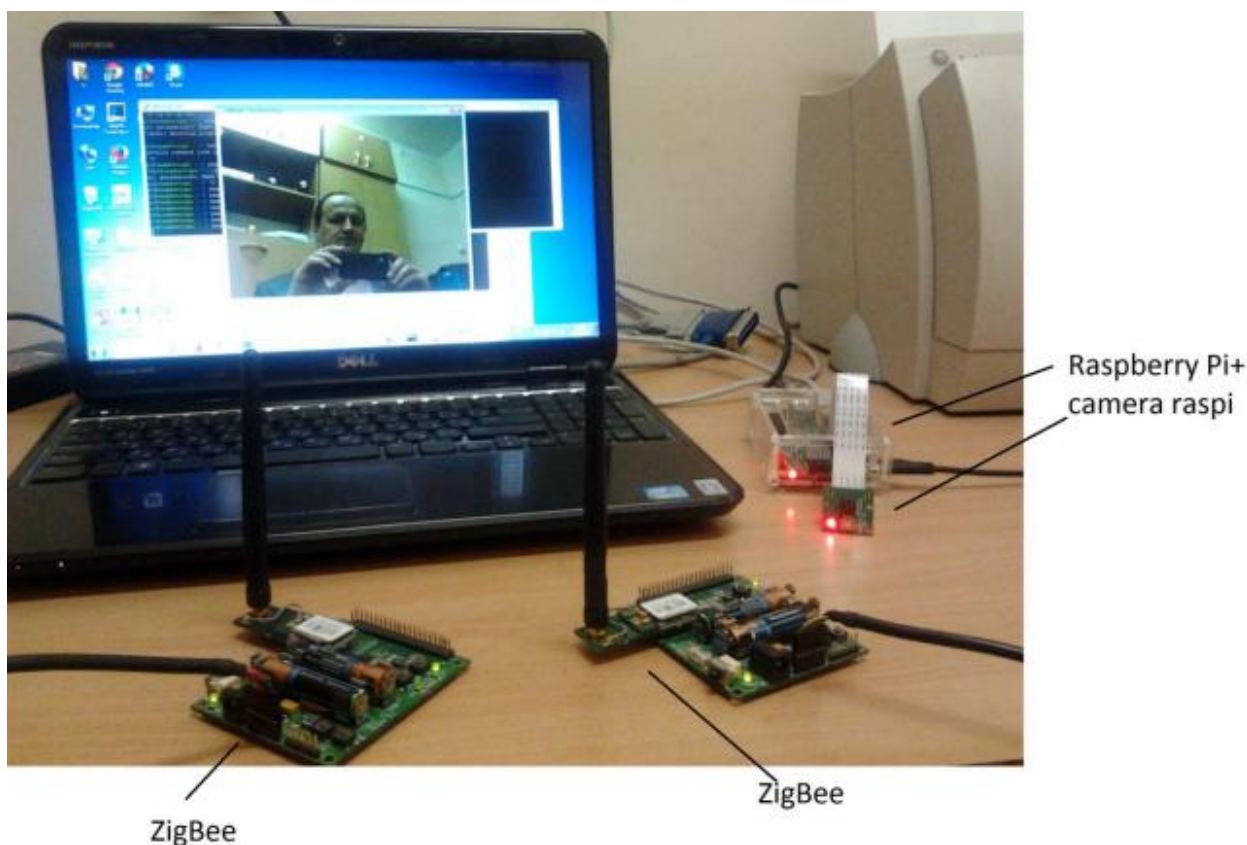


Рисунок 6.12 – Передача зображень з використанням безпроводної сенсорної мережа

Структурна схема розробленого вузла БМСМ наведена на рисунку 6.13. Використання модуля обробки даних в структурі безпроводного модуля відеоспостереження дозволяє проводити попередню обробку мультимедійного контенту, зокрема, цифрову обробку, стиснення даних, розділення на підпакеми (рисунок 6.13). Проведений аналіз ринку сучасних одноплатних мікрокомп'ютерів показав можливість їх використання для обробки мультимедійного контенту в БМСМ (таблиця 6.3). Одноплатний комп'ютер - пристрій, зібраний на одній друкованій платі, на якій встановлені мікропроцесор, оперативна пам'ять, системи вводу-виводу та інші модулі, необхідні для функціонування комп'ютера [99].

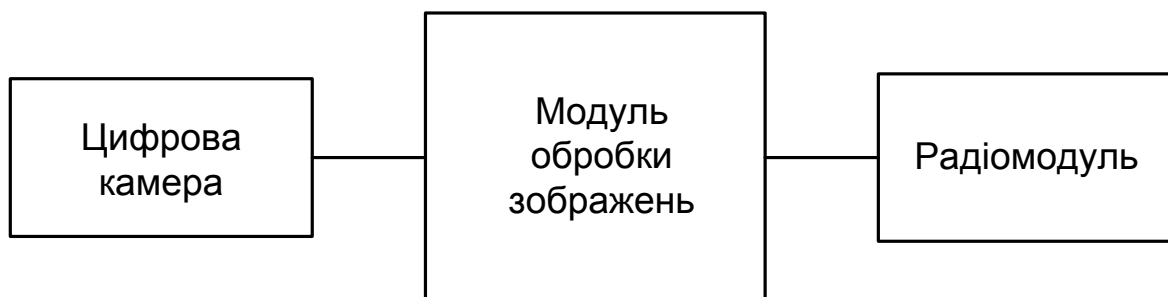


Рисунок 6.13 – Структурна схема безпроводного модуля системи відеоспостереження

Таблиця 6.3 – Параметри мікрокомп'ютерів

Назва	Процесор	ОЗП	Інтерфейси	ОС	Живлення	Розмір, мм
Edison Intel	Intel Quark, 400 МГц	512 МБ	I2C, I2S, UART, GPIO і PWM, Wi-Fi e, Bluetooth LE	Linux	3,3 В, 250 мА	32 × 24
Raspberr y Pi	ARM11, Broadcom BCM2835 , 700 МГц	512 МБ	HDMI, USB, Stereo Jack 3.5 мм, Ethernet, UART, GPIO, JTAG, SPI, I2C	Linux, Firefox OS, Android	5 В, 700 мА	86 × 54
Cubie- Board	Allwinner A10, 1 ГГц	1 ГБ, DDR3	Ethernet, 2xUSB, SATA, I2C, SPI, RGB/LVDS, ADC, CVBS, VGA та ін.	Android, Linux	5 В, 2А	100×60

Серед наявних на ринку мікрокомп'ютерів (таблиця 6.3) необхідно виділити дві платформи – Intel®Edison та Raspberry Pi. Edison –

мікрокомп'ютер на базі технології Intel®Quark, за розмірами рівний карті пам'яті SD. Перевагою даного модуля є вбудовані технології безпроводної передачі даних з низьким енергоспоживанням Wi-Fi e (low power), Bluetooth LE, що забезпечує можливість роботи від автономного або відновлюваного (сонячні елементи) джерела живлення. Перевагою мікрокомп'ютера Raspberry Pi є вбудований інтерфейс для підключення камери, потужність процесора та ціна (рисунок 6.14). В якості камери використано 5 МП камеру від команди розробників комп'ютера Raspberry Pi, яка підключається шлейфом до одного зі спеціальних роз'ємів CSI на платі. Камера має фіксований фокус і надає можливість записувати відео в трьох режимах: 1080p (30 кадрів), 720 p (60 кадрів) і VGA (60 або 90 кадрів), або робити фотографії з роздільною здатністю 2592 x 1944 пікселів. Розміри плати, яка використовується, 25 x 20 x 9 мм та контролер - BCM2835.

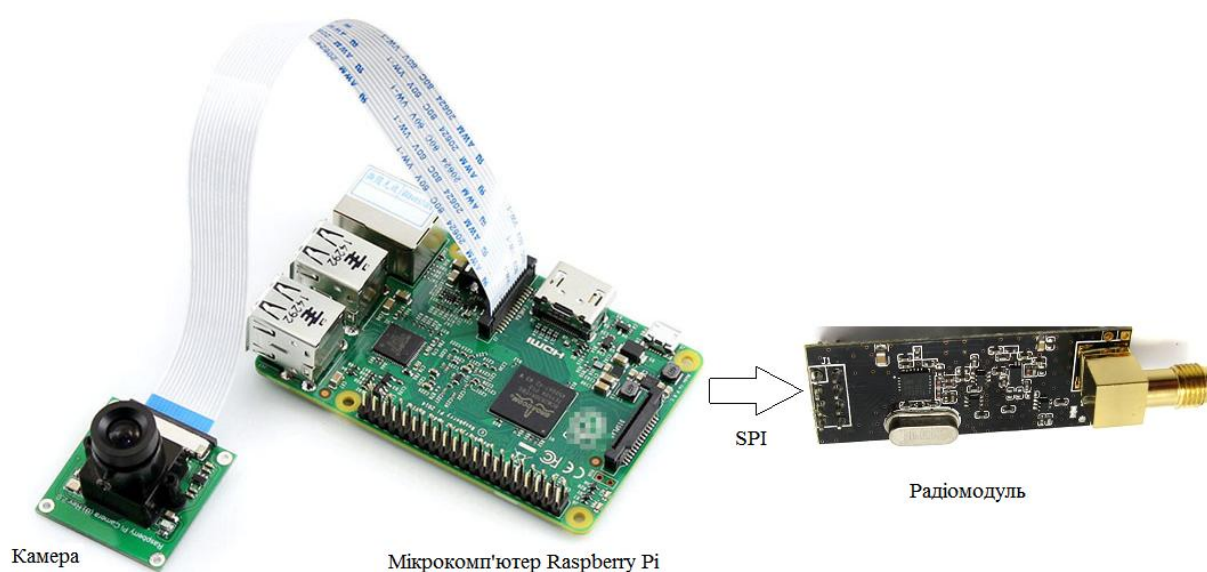


Рисунок 6.14 – Схема мультимедійного вузла БМСМ на базі мікрокомп'ютера

Структура вузла БМСМ на базі реалізованого на ПЛІС спецпроцесора наведена на рисунку 6.15. Вузол БМСМ складається з 5-мегапіксельної

цифрової камери, яка підтримує зображення розміром 2592 x 1944 активних пікселів, ПЛІС та радіомодуля. На виході отримуємо зображення в форматі RGB і частотою до 15 кадрів в секунду. Радіомодуль для зв'язку вузлів між собою працює на частоті 2.4ГГц, швидкість до 2 Мбіт, управляється по інтерфейсу SPI, живлення 3.3 В.

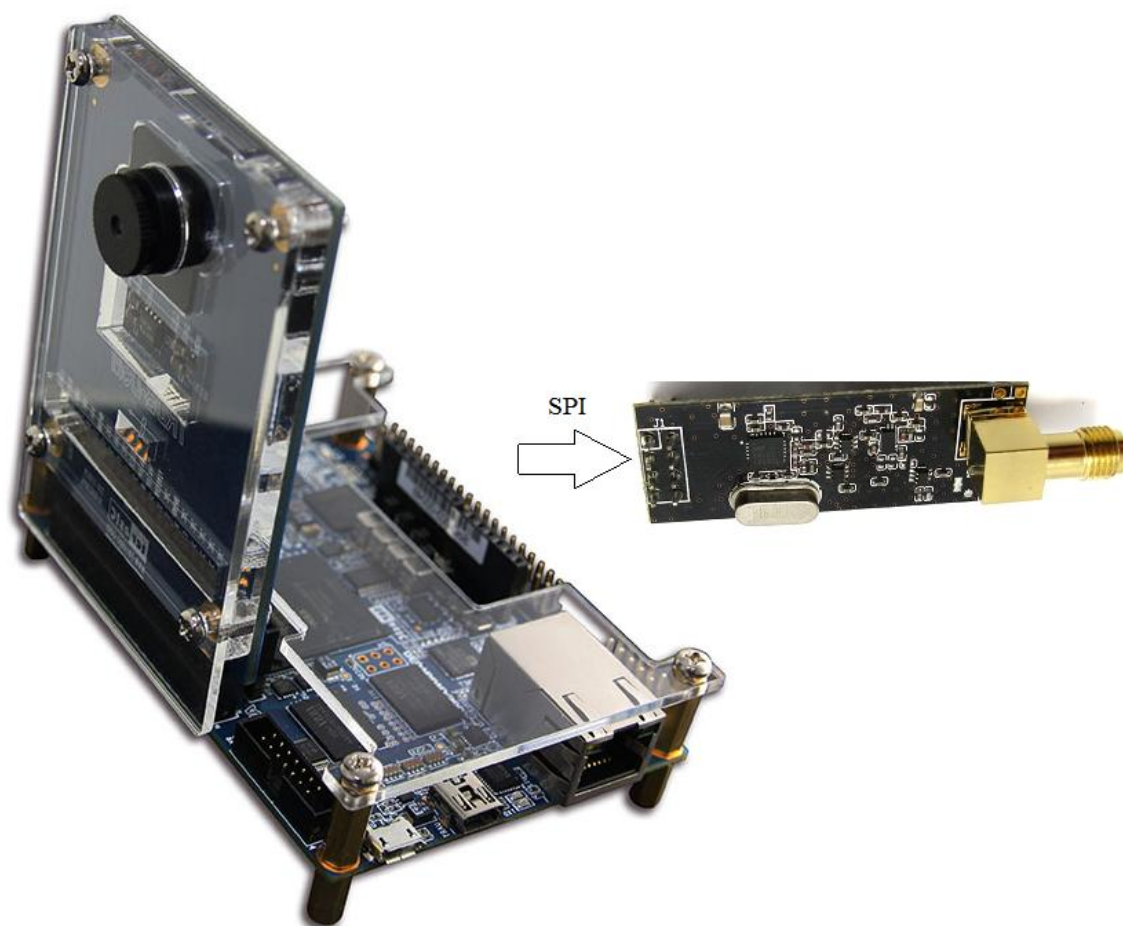


Рисунок 6.15 – Схема мультимедійного вузла БМСМ на базі спецпроцесора на ПЛІС

Характеристики вузла: швидкість до 2 Мбіт (налаштовується 2, 1, 0,25 Мбіт/ с); 126 каналів зв'язку, підключення один до багатьох, стрибкоподібна зміна частоти; апаратний контроль помилок, контроль адресації багатоточкового зв'язку; низьке енергоспоживання: 1,9 – 3,6 В, в режимі сну – 1мкА; зовнішня або вбудована антена на 2,4 ГГц; модуляція GFSK [99].

Кодер виконує перетворення двійкового коду зображення в код системи залишкових класів з використанням розширеної системи модулів, що дозволяє виявляти та виправляти помилки.

Алгоритми стиснення даних для використання в БМСМ повинні задовольняти такі вимоги [81]:

- 1) забезпечувати заданий рівень стиснення;
- 2) низьке енергоспоживання;
- 3) низька обчислювальна складність;
- 4) низькі вимоги до пам'яті;
- 5) ефективна реалізацію пристроїв кодування/декодування.

Проте алгоритми повинні також мати високу ефективність, оскільки вузли мають обмеження в пропускну здатності каналів зв'язку і високі затрати енергії на передачу даних. Однак ці вимоги суперечливі, оскільки більш складні методи стиснення зазвичай забезпечують більший коефіцієнт стиснення і, відповідно, складніші методи стиснення призводять до більшого енергоспоживання.

При передачі мультимедійних даних в БСМ, крім обмеження розміру пам'яті відеосенсорів, необхідно також враховувати обмеження пропускну здатності каналів зв'язку.

6.3.1 Пристрій для перетворення зображення в систему залишкових класів

Для реалізації запропонованого в пункті 2.5 методу перетворення зображень розроблено пристрій, який перетворює двійковий код зображення в систему залишкових класів з розширеною системою модулів [52, 65].

Перетворення восьмибітного двійкового коду в код системи залишкових класів реалізовано на основі методу безпосереднього додавання та запропонованої в [52, 202] структури пристрою

$$A = \sum_{j=0}^k a_j \cdot 2^j = \sum_{j=0}^k A_{ij} \pmod{p_i},$$

$$A_{ij} \equiv (a_i \cdot 2^j) \pmod{p_i},$$

$$A_{ij} = 0, \text{ при } a_j = 0,$$

$$A_{ij} = 2^j \pmod{p_i}, \text{ при } a_j = 1 \quad (6.6)$$

Структурна схема розробленого пристрою зображена на рисунку 6.16 [99]. Для розділення зображення на частини вибрані модулі: $p_1 = 5$, $p_2 = 7$, $p_3 = 8$, $p_4 = 11$, $p_5 = 13$, з яких $p_1 - p_3$ – інформаційні, p_4 і p_5 – перевірочні. Двійковий код зображення (8 розрядів) надходить на вхідний регістр 1. Зчитана з регістра інформація надходить на входи шифраторів 2. Зчитана з регістра інформація надходить на входи шифраторів 2.

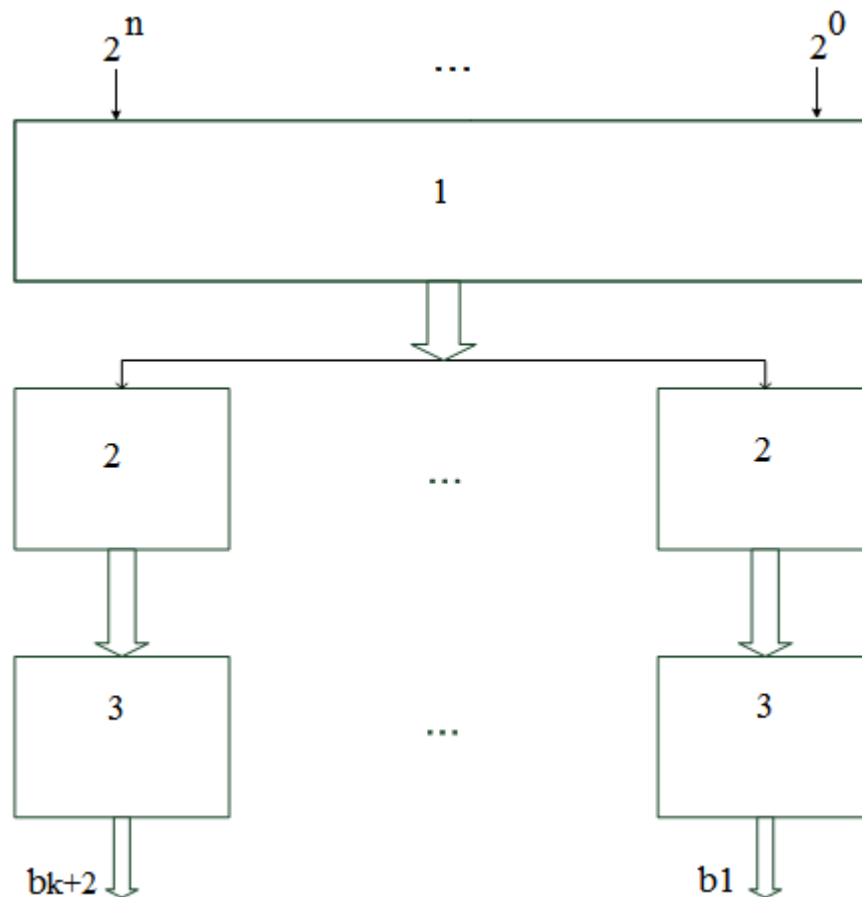


Рисунок 6.16 – Структурна схема пристрою перетворення двійкового коду в код системи залишкових класів: 1 – вхідний регістр; 2 – неповні шифратори; 3 – пірамідальні суматори по заданому модулю

При наявності в j – му розряді регістра одиниці на виході шифраторів формується двійковий код, який відповідає коефіцієнтам згідно виразу $a_{ij} = 2^j \bmod p_i$. З виходів шифраторів двійковий код надходить на вхід суматорів 3 по модулях p_i .

На вхід шифратора поступають розряди двійкового коду. На виході шифратора формуються коефіцієнти згідно формули (6.6) та таблиці 6.4.

Таблиця 6.4 – Коефіцієнти на виході шифратора при одиничному інформаційному біті на вході

	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0
$p_1=5$	3	4	2	1	3	4	2	1
$P_2=7$	2	1	4	2	1	4	2	1
$P_3=8$	0	0	0	0	0	4	2	1
$P_4=11$	7	9	10	5	8	4	2	1
$P_5=13$	11	12	6	3	8	4	2	1

Робота шифратора описується системою логічних рівнянь:

– для модуля $p_1 = 5$:

$$q_{0_0} = a_0 \wedge x;$$

$$q_{1_0} = a_1 \wedge \bar{x};$$

$$q_{1_1} = a_1 \wedge x;$$

$$q_{2_0} = a_2 \wedge \bar{x};$$

$$q_{2_1} = a_2 \wedge x;$$

$$q_{2_2} = a_2 \wedge x,$$

....

$$q_{j_i} = a_j \wedge x$$

...

$$q_{14_0} = a_{14} \wedge \bar{x};$$

$$q_{14_1} = a_{14} \wedge \bar{x};$$

$$q_{14_2} = a_{14} \wedge x;$$

$$q_{15_0} = a_{15} \wedge x.$$

де a_j – значення j -го розряду двійкового числа, $x=1$, $\bar{x}=0$ – константи.

Аналогічно записуються рівняння, які описують роботу шифратора для модулів $p_2 = 7$, $p_3 = 8$, $p_4 = 11$, $p_5 = 13$.

Функціональна схема неповного шифратора для модуля $p_1 = 5$ зображена на рисунку 6.17.

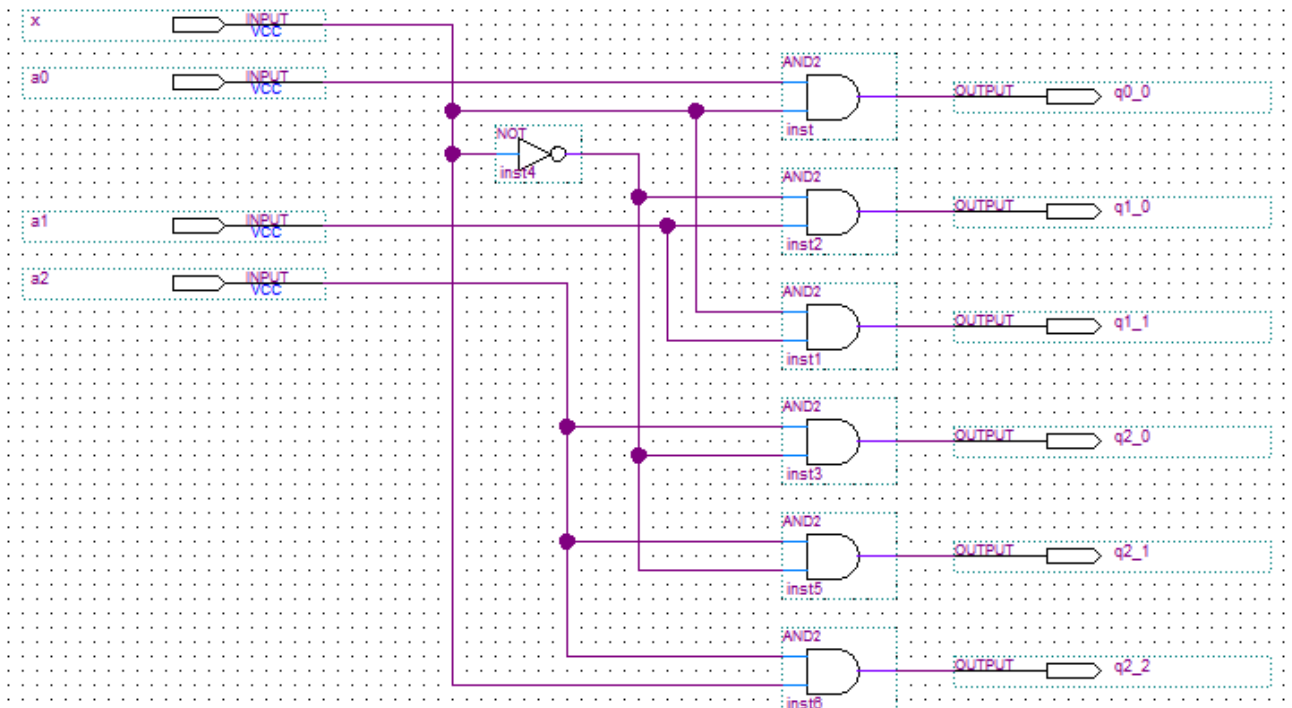


Рисунок 6.17 – Функціональна схема неповного шифратора формування коефіцієнтів по модулю $p_1 = 5$

Додавання коефіцієнтів, які надходять із шифратора, реалізовано в пірамідальному суматорі за відповідним модулем, що дозволило підвищити швидкодню перетворення (рисунок 6.18).

Розроблений пристрій перетворення зображення в СЗК та часові діаграми, які підтверджують його коректну роботу, наведені на рисунках 6.19, 6.20 відповідно.

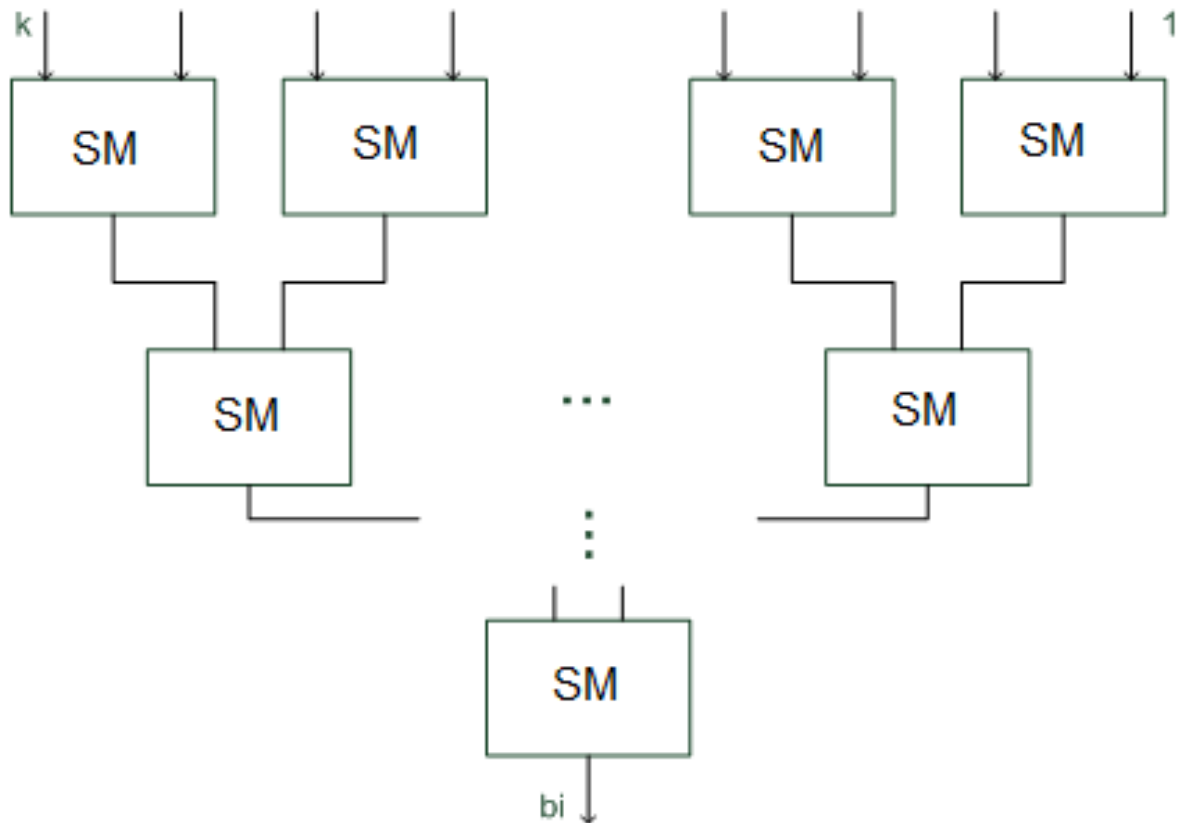


Рисунок 6.18 – Структурна схема пірамідального суматора:

SM – суматор по модулю p_i ; b_i – залишок по модулю p_i

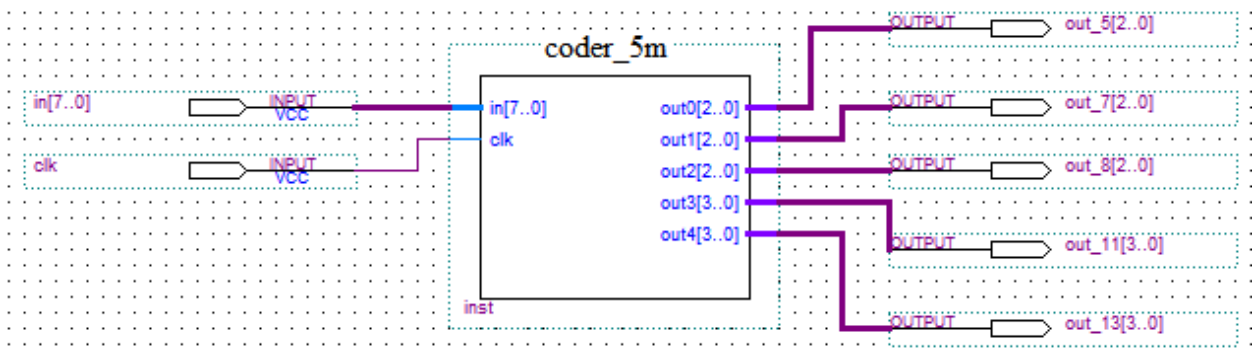


Рисунок 6.19– Пристрій перетворення зображення в СЗК

На вхід пристрою може поступати двійковий код як стиснутих даних (формат jpg та ін.), так і зображень, поданих за допомогою кольорових моделей RGB, YCbCr та інших. Отримані залишки передаються по незалежних маршрутах з використанням багатошляхової маршрутизації.

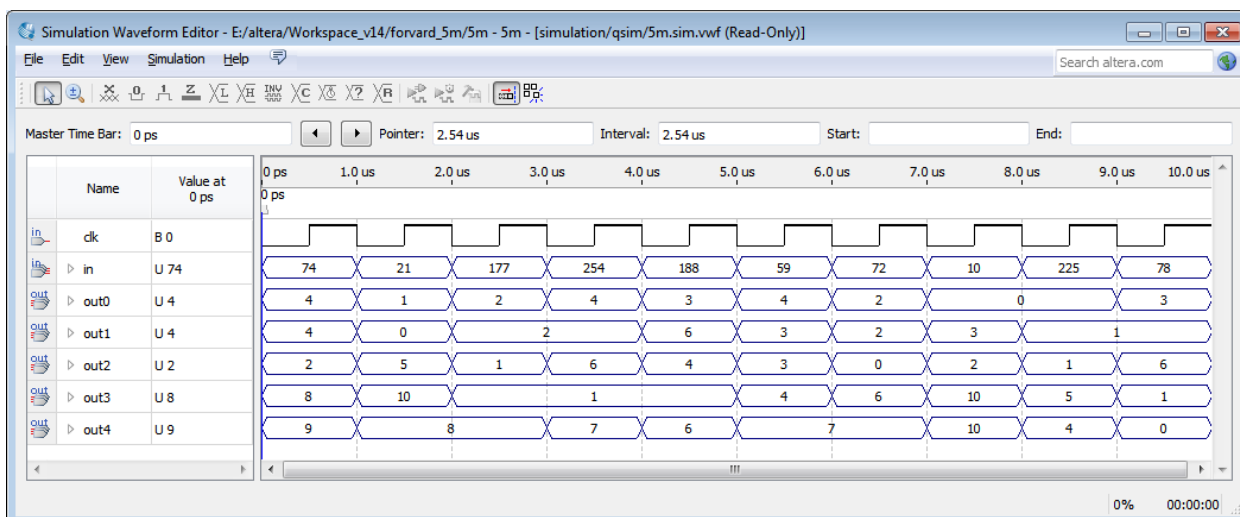


Рисунок 6.20 – Часові діаграми роботи пристрою обробки зображення

Таким чином, розроблений пристрій перетворення зображення в СЗК з розширеною системою модулів забезпечує зменшення апаратних затрат в 2,6 рази та підвищення швидкодії на 47%, порівняно з відомими схемами перетворення, за рахунок використання неповних шифраторів та пірамідальних суматорів за відповідними модулями. Пристрій реалізовано на мікросхемі 5CSEMA4U23C6N серії Cyclone® V SE, фірми ALTERA, час перетворення 8 - бітного двійкового коду становить 9,923 нс, відповідно, час перетворення одного кадру зображення (1024 * 768) приблизно дорівнює 7.8 мс.

6.4 Підвищення надійності передачі даних у системі дистанційного керування технологічними процесами „СТРІЛА-М”

Запропоновані в розділі 3 (пункт 3.5) модулярні коректуючі коди використані для підвищення надійності передачі даних у системі дистанційного керування технологічними процесами “СТРІЛА-М” [26].

Система “СТРІЛА-М” забезпечує керування до 127 контрольованими пунктами та зняття даних з них (рисунок 6.21). Для кожного з контрольованих пунктів (КП) комплекс забезпечує:

- 1) телекерування (ТК) – до 32 об’єктів;
- 2) телесигналізація (ТС) – до 64 каналів;
- 3) телевимірювання (ТВ) – до 16 каналів;
- 4) зняття даних з електронних лічильників електроенергії – до 32 шт.
- 5) При необхідності число каналів ТК, ТС, ТВ і кількість лічильників електроенергії може бути збільшено, крок нарощування каналів ТК - 8, ТС - 32, ТВ - 16.

б) Керування об’єктами контрольованого пункту здійснюється з персонального комп’ютера диспетчерського пункту керування з відображенням на екрані дисплея станів телесигналізації і параметрів телевимірювання об’єктів КП у вигляді мнемосхеми і текстової інформації згідно виконавчих схем первинних з’єднань. Підключення обладнання ДПК комплексу до ПК здійснюється через послідовний порт передачі даних (RS 232) або USB-2.

В основу роботи системи “СТРІЛА-М” покладений принцип «ведучий-ведений». У якості «ведучого» виступає апаратура ДПК, а в якості «веденого» – апаратура АККП. Згідно цього принципу, апаратура АККП працює в автоматичному режимі і постійно відслідковує стан телемеханічного обладнання контрольованого пункту (КП).

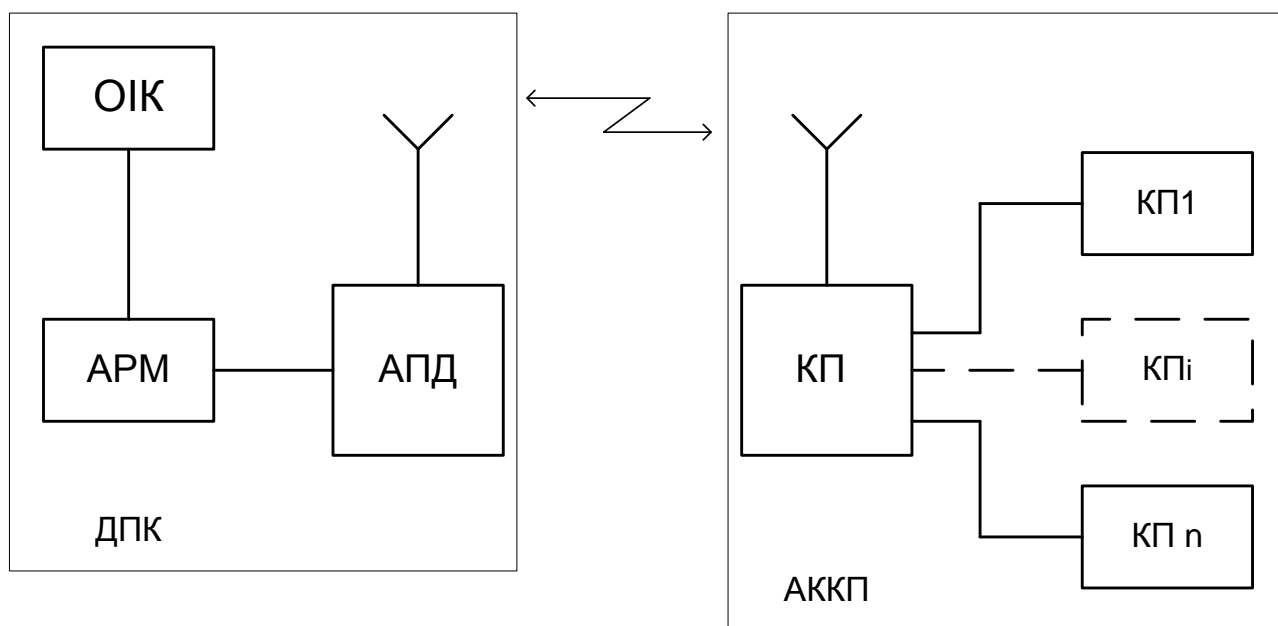


Рисунок 6.21 – Загальна структура системи дистанційного керування технологічними процесами “СТРІЛА-М”: ОІК – оперативно - інформаційний комплекс; АРМ - автоматизоване робоче місце оператора; АПД – апаратура передачі даних; КР – контрольований пункт; ДПК – диспетчерський пункт керування; АККП – апаратура керування контрольованого пункту

Ініціатором обміну інформації між ДПК і АККП виступає апаратура ДПК, тобто апаратура ДПК постійно, по чергово в циклі опитує АККП і знімає з них інформацію. В разі виникнення на АККП позаштатної ситуації, відбувається самочинна зміна стану телесигналізації (ТС), апаратура фіксує цю інформацію і при черговому сеансі зв’язку відправляє її на ДПК з мітками часу, а при штатній роботі АККП – видає коротку кодограму про відсутність змін параметрів роботи контрольованого пункту; при наявності ще одного (резервного) каналу зв’язку інформація про позаштатну роботу обладнання КР передається в сторону ДПК в момент її виявлення [26].

Для передачі даних система використовує один основний канал зв’язку, по якому відбувається обмін інформацією в штатному режимі роботи, другий – резервний, для передачі аварійних кодограм з контрольованого пункту на ДПК. При наявності в системі більше 10

контрольованих пунктів обмін даними відбувається паралельно по двох каналах зв'язку.

На рисунку 6.22 наведено структурну схему обладнання ДПК. Шафа комунікаційна (радіомодем з блоком живлення) та персональний комп'ютер можуть розміщатися [26]:

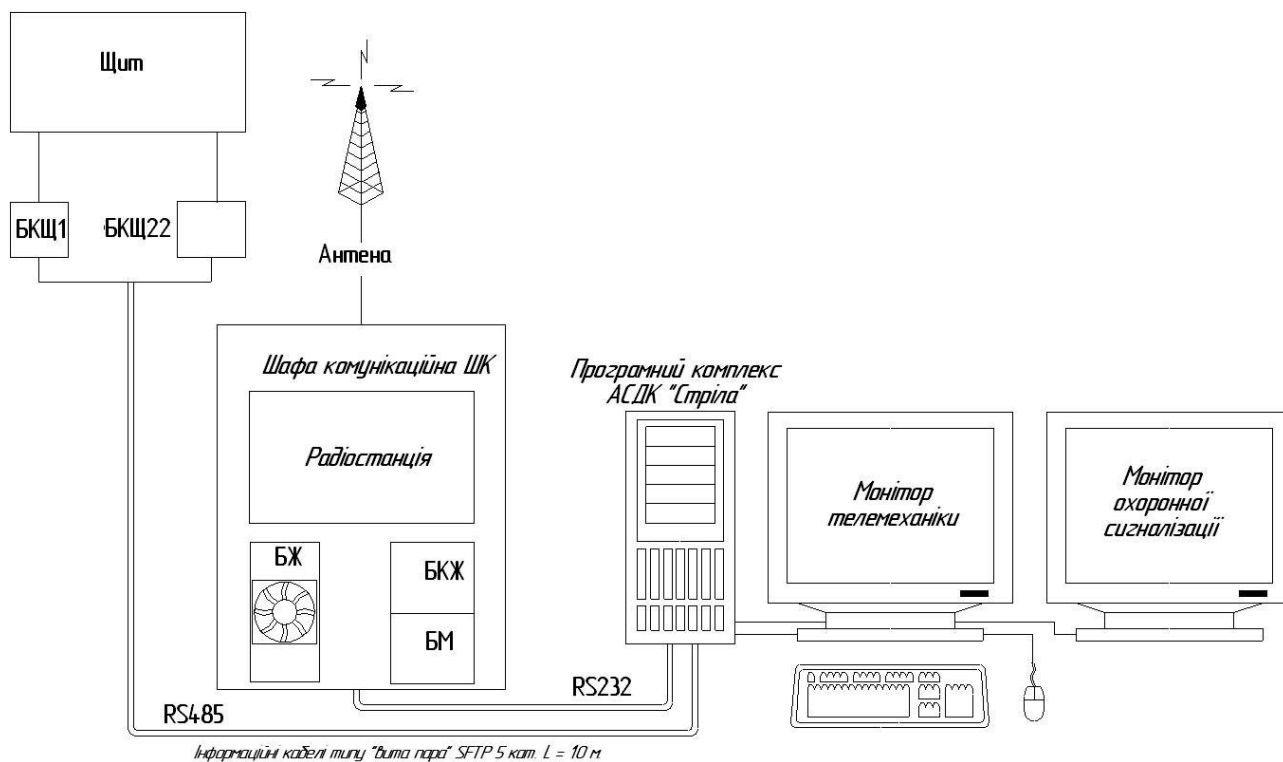


Рисунок 6.22 – Структурна схема обладнання ДПК

а) відстань між обладнанням до 10 м (кабель «вита пара» з використанням шини RS232);

б) відстань між обладнанням до 400 м (кабель «вита пара» з використанням шини RS485);

в) відстань між обладнанням більше 400 м (виділена телефонна пара, при цьому в обладнання додатково вводиться шафа комунікаційна (ШК), яка виконує функцію узгодження з виділеними телефонними каналами) (див. рисунок 6.22).

Обладнання АККП побудоване по модульному принципу (рисунок 6.23). Кожний блок має в своєму складі контролер керування, виконаний на базі мікроконтролера. Всі блоки автономно виконують свої функції

(контроль за станом сигналів телесигналізації, телевиміри і т.п.). Всі блоки об'єднані між собою внутрішньою шиною RS485. Керування процесами АККП виконує блок контролер збору даних, який по шині RS232 під'єднаний до радіомодему.

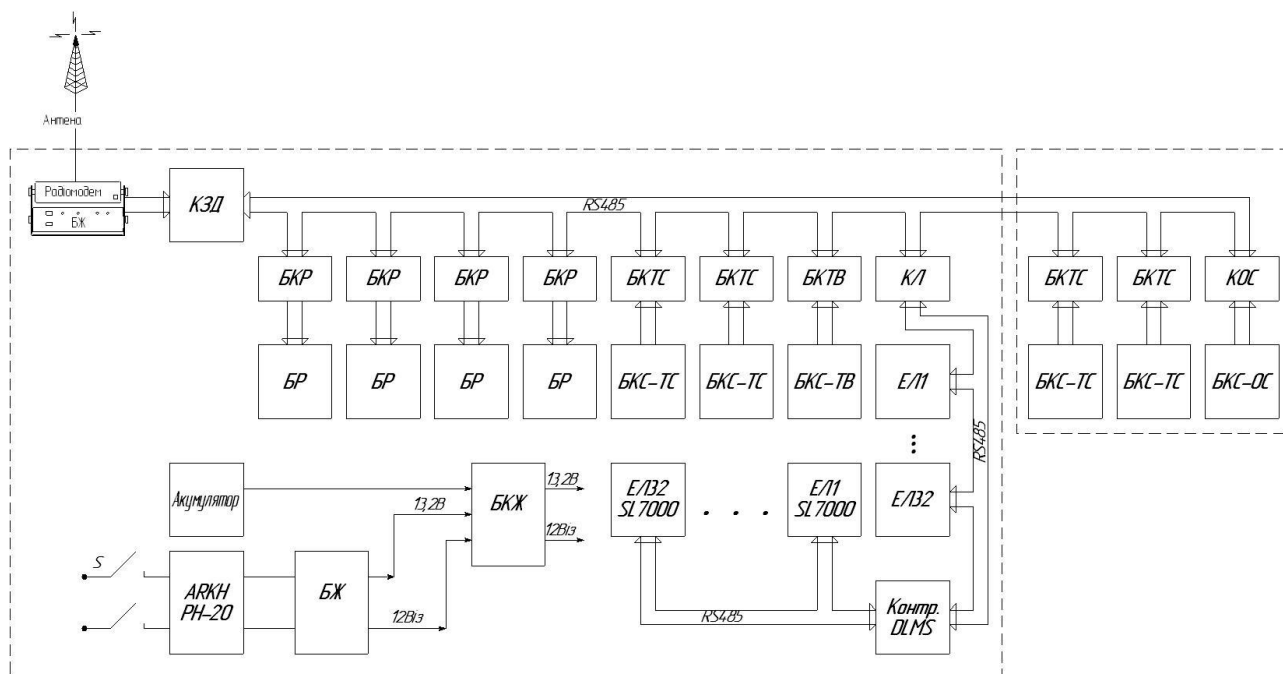


Рисунок 6.23 – Структурна схема обладнання контрольованого пункту (АККП): АРКН - автоматичне реле контролю напруги; БЖ - блок живлення; КЗД - контролер збору даних; БКР - блок керування реле; БР - блок реле; БКТС - блок керування телесигналізацією; БКС-ТС - блок комутації сигналів ТС; БКТВ - блок керування телевимірами; БКС-ТВ - блок комутації сигналів ТВ; КЛ - контролер лічильників електроенергії; ЕЛ – лічильник електроенергії; КОС – контролер охоронної сигналізації; БКС-ОС - блок комутації сигналів ОС

Інформація через радіомодем поступає на блок КЗД. На блоці КЗД інформація обробляється, в результаті чого КЗД видає команду на відповідний блок АККП для виконання. Після виконання команди блок КЗД отримує інформацію від відповідного блоку, формує пакет інформації, і через радіомодем передає її на апаратуру ДПК.

Для зв'язку між диспетчерським і контрольованими пунктами використовуються пристрої зв'язку на базі вузькосмугових радіоканалів

неправильний. І тільки при позитивних результатах перевірок кадр видається користувачеві.

Показник «достовірність переданих даних» в системах телемеханіки характеризується ймовірністю появи невиявлених помилок при ймовірності спотворення біту 10^{-4} і нормується відповідно до таблиці 6.5 [155, 10].

Таблиця 6.5 – Класи достовірності передачі даних

Клас достовірності даних	Ймовірність появи невиявлених помилок
I1	$\leq 10^{-6}$
I2	$\leq 10^{-10}$
I3	$\leq 10^{-14}$

Згідно [3] частота невиявлених помилок для даних користувача повинна відповідати класу достовірності даних I2, для команд – класу достовірності даних I3.

Для забезпечення необхідного класу достовірності передачі даних застосовують наступні підходи [154]:

- контроль якості сигналу, що передається;
- передача даних з використанням коректуючих кодів;
- ефективні засоби виявлення помилок;
- синхронізація блоків даних;
- повторна передача даних;
- оцінка стану лінії зв'язку;
- інформаційний зворотній зв'язок.

Для підвищення достовірності передачі даних в системі “СТРІЛА-М” та забезпечення сумісності із вже впровадженими на об'єктах електроенергетики системами (на даний час введено в експлуатацію більше 100 комплектів) використано запропоновані в розділі 3 модулярні коректуючі

коди та розроблено модифіковану структуру формату кадру передачі даних (рисунок 6.25).

Враховуючи, що максимальна довжина повідомлень в системі “СТРІЛА-М” становить $L \leq 32$ байти, використано модулярний коректуючий код з параметрами: розрядність символів $m = 8$ біт, кількість символів $k = 8$, відповідно, повідомлення розділяється на чотири блоки даних. Після кожного блоку даних, який складається з 8-ми інформаційних символів, обчислюються і передаються перевіірочні символи CS1 – CS4 (рисунок 6.25).

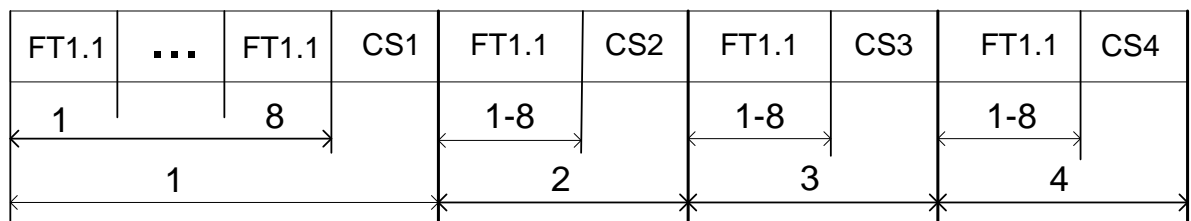


Рисунок 6.25 – Структура пакету передачі даних з використанням модулярних коректуючих кодів

Для передачі перевіірочні символи (16 біт) розділяються на дві частини по 8 біт і передаються двома кадрами формату FT1.1 (рисунок 6.26).

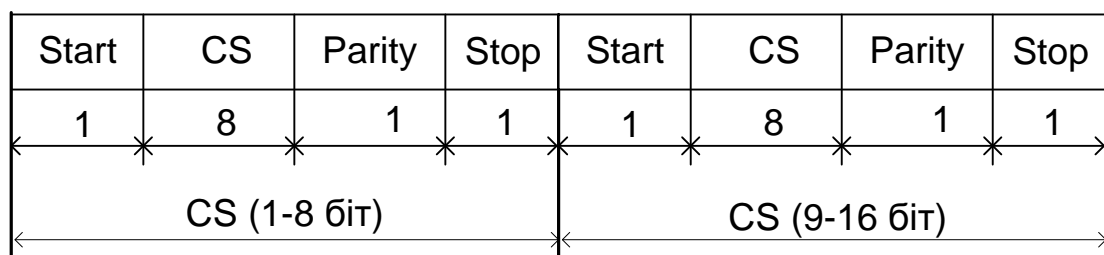


Рисунок 6.26 – Передача перевіірочних символів

Для визначення допустимої імовірності помилок при передачі даних розрахуємо залежність середнього часу між появою невиявлених помилок в повідомленні і ймовірністю появи невиявлених помилок [17, 155]:

$$T = \frac{n}{C \cdot P},$$

де n – довжина повідомлення в бітах;

C – швидкість передачі, біт/с;

P – ймовірність появи невиявлених помилок.

Отже час між невиявленими помилками, при передачі повідомлень зі швидкістю 2400 біт/с становить: 16,95 роки при розмірі повідомлень 128 біти та 270 років при розмірі повідомлень 2048 бітів (рисунок 6.27). Відповідно, для забезпечення надійної роботи системи “СТРІЛА-М” необхідно, щоб ймовірність появи невиявлених помилок становила $R = 10^{-10}$.

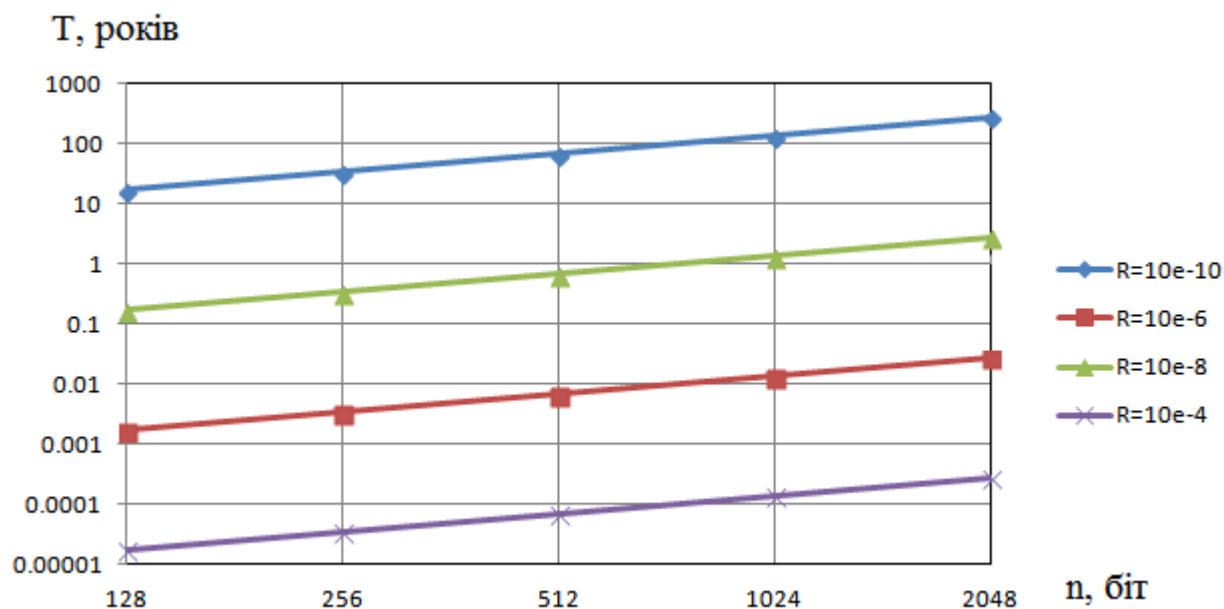


Рисунок 6.27 – Час між появою невиявлених помилок в залежності від обсягу повідомлень при різних значеннях ймовірності появи помилок

Для обґрунтування переваг застосування коректуючих кодів розрахуємо ймовірність появи помилок в повідомленні до і після кодування.

Ймовірність символічних помилок в каналі без кодування [60]:

$$p_u = Q \cdot \sqrt{2 \cdot E_b / N_0},$$

з кодуванням

$$p_c = Q \cdot \sqrt{2 \cdot E_c / N_0},$$

де E_b/N_0 – відношення енергії біту до спектральної густини потужності шуму;

E_c/N_0 – відношення енергії кодованого біту до спектральної густини потужності шуму;

$$Q(x) \text{ – інтеграл помилок Гауса, } Q(x) \approx \frac{1}{x \cdot \sqrt{2 \cdot \pi}} \cdot \exp\left(\frac{-x^2}{2}\right).$$

Без кодування:

$$\frac{E_b}{N_0} = \frac{P_r}{N_0} \cdot \left(\frac{1}{R}\right) = 14.58,$$

$$\text{де } R = 2400 \text{ біт/с, } \frac{P_r}{N_0} = 35000.$$

Отже,

$$p_u = Q \cdot \left(\sqrt{\frac{2 \cdot E_b}{N_0}}\right) = Q \cdot (\sqrt{29,17}) = 1.29 \cdot 10^{-5}.$$

Ймовірність того, що некодований блок повідомлення буде прийнятий з помилкою [60]:

$$P_M^u = 1 - (1 - p_u)^k = 1.03 \cdot 10^{-4}.$$

З кодуванням.

Швидкість передачі кодованих символів дорівнює:

$$R_c = 2400 \cdot \frac{10}{8} = 3000 \text{ біт/с,}$$

і

$$\frac{E_c}{N_0} = \frac{P_r}{N_0} \cdot \left(\frac{1}{R_c} \right) = 11.67.$$

Ймовірність помилки в символі [60]:

$$p_c = Q \cdot \left(\sqrt{\frac{2 \cdot E_c}{N_0}} \right) = Q \cdot (\sqrt{11.67}) = 1.15 \cdot 10^{-5}.$$

Ймовірність появи помилки в закодованому символі визначимо через ймовірність появи помилки в каналному символі p_c [60]:

$$P_M^c \approx \frac{1}{2^m - 1} \cdot \sum_{j=t+1}^{2^m-1} j \cdot \binom{2^m - 1}{j} \cdot p_c (1 - p_c)^{2^m - 1 - j}, \quad (6.7)$$

де m – розрядність символів;

t – кількість помилок, які може виправити код.

Підставивши в (6.7) числові значення, одержимо:

$$P_M^c \approx 3.37 \cdot 10^{-8}.$$

Таким чином, вдосконалення структури кадру FT1.2 за рахунок використання модулярних коректуючих кодів, які виправляють помилку в одному символі, дозволило зменшити ймовірність помилки в повідомленні приблизно в 3060 разів.

Висновки до шостого розділу

1. Проведено дослідження надійності роботи БСМ та розраховано затрати енергії при використанні різних схем контролю помилок. Отримані результати показали, що коректуючі коди з декодуванням в проміжних вузлах необхідно використовувати при рівні завад ($BER > 10^{-3}$) і вище, а метод ретрансляції пакетів (з декодуванням в кінцевому вузлі) має переваги при рівні завад ($BER < 10^{-3}$) та передачі даних чутливих до затримок.

2. Розроблено методику тестування каналів зв'язку безпроводних сенсорних мереж, яка дозволяє вибирати необхідну схему контролю помилок та параметри коректуючого коду в залежності від стану каналу зв'язку. Проведено експериментальне тестування надійності передачі даних в БСМ, яке показало необхідність врахування впливу обладнання, що працює в діапазоні частот 2.4 ГГц (Wi-fi, Bluetooth та інше).

3. Розроблено топологію безпроводної мультимедійної сенсорної мережі системи відеоспостереження об'єктів та структуру безпроводного модуля. Реалізація на ПЛІС блоку обробки даних в структурі безпроводного модуля відеоспостереження дозволяє проводити попередню обробку мультимедійного контенту, зокрема, розділення на підпакети та стиснення даних.

4. Розроблено пристрій кодування зображення в СЗК з розширеною системою модулів, який забезпечує зменшення апаратних затрат в 2,6 рази та підвищення швидкодії на 47%, порівняно з відомими схемами перетворення, за рахунок використання неповних шифраторів та пірамідальних суматорів за відповідними модулями.

5. Вдосконалено структуру кадру протоколу передачі даних в системі дистанційного керування технологічними процесами „СТРІЛА-М”, що дозволило зменшити ймовірність помилки в повідомленні приблизно в 3060 разів за рахунок використання модулярних коректуючих кодів.

ВИСНОВКИ

У дисертаційній роботі вирішено актуальну науково-прикладну проблему – підвищення ефективності функціонування безпроводних сенсорних мереж шляхом розвитку теоретичних основ та засобів перетворення, обробки і передачі даних на базі системи залишкових класів та модулярної арифметики. При цьому отримано такі основні теоретичні й практичні результати і наукові висновки:

1. Проведено аналіз стану розробки та сфери використання БСМ на їхній основі, сформульовано вимоги до функцій і характеристик безпроводних вузлів та мереж. Виділено переваги й існуючі обмеження, які стримують широке використання БСМ. Розроблено класифікацію БСМ. Вказано перспективи використання і виділено існуючі проблеми при створенні безпроводних мультимедійних сенсорних мереж. Досліджено перспективні методи підвищення надійності та корисної пропускної здатності безпроводних сенсорних мереж.

2. Запропоновано метод мережного кодування даних на основі системи залишкових класів, який підвищує загальну пропускну здатність мережі за рахунок поділу повідомлення на частини та передачі їх різними маршрутами. Вибір взаємно простих модулів різної розрядності забезпечує рівномірний розподіл трафіка між маршрутами мережі, оскільки розрядність залишків, які передаються спільним маршрутом, приблизно дорівнює розрядності залишків, які передаються різними маршрутами. Розроблено протокол передачі даних на основі мережного кодування даних, який забезпечує підвищення пропускної здатності мережі, порівняно з протоколом стандарту IEEE 802.15.4, в середньому на 54% при збільшенні затримки передачі пакетів на 9% (0,001 с).

3. Розроблено модифікований метод стрибкоподібної зміни частоти, який забезпечує підвищення надійності передачі даних на фізичному рівні

безпроводних сенсорних мереж та захист від спланованих завад за рахунок поділу повідомлення на систему взаємно простих модулів та передачі отриманих залишків за допомогою набору несучих частот, вибір яких здійснюється за псевдовипадковим законом.

4. Запропоновано метод перетворення й обробки зображень, який забезпечує підвищення швидкодії обробки зображення в 2–3 рази за рахунок поділу зображення на модулі системи залишкових класів і паралельного стиснення отриманих залишків. Стиснення отриманих залишків методом арифметичного кодування забезпечує коефіцієнт стиснення без втрат для зображень від 13% до 24% (залежно від формату зображення) та від 54% до 69% для текстових файлів.

5. Розроблено модулярні коректуючі коди, які обробляють вхідні дані, подані двійковим кодом, і забезпечують гнучкий вибір розрядності та кількості інформаційних символів. Наявність одного перевірного символу забезпечує 100% виявлення і виправлення помилок в одному інформаційному символі.

6. Розроблено метод виправлення багатократних помилок на основі модулярного коректуючого коду, в якому перевірочні символи формуються з використанням наборів різних коефіцієнтів та спільного модуля. Ці коди забезпечують виправлення помилок в інформаційних символах, кількість яких дорівнює кількості перевірочних символів, при цьому час формування перевірочних символів приблизно в 5 разів менший порівняно з коректуючими кодами СЗК і кодом Ріда – Соломона (127, 87). Апаратні затрати на реалізацію кодування у середньому в 16 разів менші порівняно з коректуючими кодами СЗК. Використання коректуючого коду в БСМ зменшує відношення E_b/N_0 на 3 дБ при збереженні заданої надійності передачі даних. Розміщення перевірочних символів у кінці поля даних протоколу забезпечує сумісність з мережами стандарту IEEE 802.15.4.

7. Запропоновано метод виявлення та виправлення помилок при виконанні арифметичних операцій (додавання, віднімання, множення) на основі модулярних коректуючих кодів, який дає змогу виявляти помилки у двох десяткових розрядах та виправляти їх в одному десятковому розряді при використанні одного контрольного символу. Розроблено структурну схему блоку виявлення та виправлення помилок при виконанні арифметичних операцій.

8. Розроблено метод підвищення надійності передачі даних у безпроводних сенсорних мережах на основі модулярних коректуючих кодів і декомпозиції даних, який забезпечує виправлення пакетів помилок довжиною k (k – кількість інформаційних символів) при одному перевірочному символі. Використання коректуючого коду з двома перевірочними символами забезпечує виправлення пакетів помилок довжиною $2 \cdot k$, оскільки не більше, ніж два пошкоджених символи будуть міститися в одному рядку.

9. Розроблено концепцію побудови безпроводних сенсорних мереж, яка базується на принципах колективного інтелекту та розподіленого процесора і забезпечує виділення частини обчислювальних ресурсів безпроводних сенсорів для функціонування розподіленого процесора та високу живучість мережі за рахунок резервування важливих сегментів програмного забезпечення розподіленого процесора у безпроводних вузлах і децентралізованого управління мережею.

10. Розроблено структури та реалізовано на ПЛІС кодер / декодер, який виявляє та виправляє однократні помилки на основі модулярного коректуючого коду. Апаратні затрати при реалізації на ПЛІС фірми “Altera”, серії MAX II становлять: для кодера – 58 логічних елементів, максимальна затримка формування коректуючого символу становить 42,74 нс; декодера – 1108 логічних елементів, максимальна затримка формування вихідних сигналів становить 62,6 нс.

11. Розроблено структуру пристрою виявлення і виправлення багатократних помилок на основі модулярних коректуючих кодів. Проведене оцінювання апаратних затрат при використанні різних серій ПЛІС фірми “Altera” показала, що для реалізації декодера на мікросхемі EP4CGX22CF19C6 серії Cyclone IV GX без використання вбудованих блоків необхідно 6986 логічних елементів. Розроблено генератор Verilog-коду, який автоматично генерує Verilog-код декодера модулярних коректуючих кодів за заданими параметрами, що дало змогу підвищити швидкодію розробки й уникнути помилок при написанні коду.

12. Розроблено пристрій багаторівневого перетворення даних у системі залишкових класів, який забезпечує об’єднання даних, отриманих від 16 сенсорів. Перетворення вхідних даних сенсорів, поданих у СЗК, у двійкову систему числення реалізовано на основі попарного об’єднання залишків за двома модулями, що дозволило зменшити розрядність даних у процесі виконання проміжних обчислень, а це, відповідно знизило апаратні затрати.

13. Розроблено методіку тестування каналу зв’язку безпроводної сенсорної мережі, яка дає змогу вибирати необхідну схему контролю помилок та параметри коректуючого коду залежно від стану каналу передачі даних. Визначено рівень завад, при якому доцільно з позиції затрат енергії використовувати коректуючі коди. Отримані результати показали, що використання коректуючих кодів з декодуванням у проміжних вузлах необхідно застосовувати при рівні завад $BER > 10^{-3}$ і вище, а метод ретрансляції пакетів (з декодуванням у кінцевому вузлі) має переваги при рівні завад $BER < 10^{-3}$ та передачі даних, чутливих до затримок.

14. Вдосконалено структуру кадру протоколу передачі даних у системі дистанційного керування технологічними процесами “СТРІЛА-М”, що дало змогу зменшити ймовірність помилки в повідомленні приблизно у $3 \cdot 10^3$ разів за рахунок використання модулярних коректуючих кодів.

Результати дисертаційної роботи можуть бути запропоновані для використання науковими організаціями, підприємствами і компаніями, які займаються розробкою та впровадженням безпроводних сенсорних мереж з метою підвищення ефективності їхнього функціонування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Акритас А. Основы компьютерной алгебры с приложениями: Пер.с англ.- М., Мир, 1994. - 544 с.
2. Акушский И.Я. Машинная арифметика в остаточных классах / И.Я.Акушский, Д.И.Юдицкий. – М.: Сов. Радио, 1968. – 460 с.
3. Акушский И.Я., Пак И.Т. Вопросы помехоустойчивого кодирования в непозиционном коде // Вопросы кибернетики. – 1977, Т.28. – С.36-56.
4. Алгулиев Р.М. Сенсорные сети: состояние, решения и перспективы. /Р.М.Алгулиев, Т.Х.Фаталиев, Б.С.Агаев, Т.С.Алиев // Телекоммуникации. Ежемесячный научно-технический информационно-аналитический и учебно-методический журнал. – 2007. – №4. – С. 27–32.
5. Амато Вито. Основы организации сетей Cisco, том 1. Пер. с англ. – М.: Издательский дом "Вильямс", 2004. – 512 с.
6. Багатоканальний адаптивний пристрій кодування та передавання даних на основі системи залишкових класів. Патент №105430 України: МПК(2014.01) H03M 7/00, H03M 718 (2006.01) H03M 1/00, G06F 11/08 (2006.01) G06F 11/00 / Винахідник: Яцків В.В. / Власник: Тернопільський національний економічний університет – № а201213700; заявл. 30.11.2012 р.; опубл. 12.05.2014р., Бюл. № 9.
7. Барсов В.И., Сорока Л.С., Краснобаев В.А. Методология параллельной обработки информации в модулярной системе счисления : Монография.- Х.: МОН, УИПА, 2009. – 268 с.
8. Варгаузин В. Помехоустойчивое кодирование в пакетных сетях // ТелеМультиМедиа. – 2005. – №9. – С.10-16.
9. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. – М.: Диалог-Мифи, 2002. – 384 с.

10. Вулис А., Сумительнов В. Основные положения базовых стандартов ГОСТ Р МЭК на устройства и системы телемеханики. Часть 1 // Современные технологии автоматизации. – 2007. – № 4. – С.76-84.

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

12. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. – Москва: Техносфера, 2006. – 616 с.

13. Дж. Миано. Форматы и алгоритмы сжатия изображений в действии. Учеб. Пособ. – М.: Издательство Триумф, 2003 – 336 с.

14. Додонов А.Г., Ландэ Д.В. Живучесть информационных систем. – К.: Наук. думка, 2011. – 256 с.

15. Дрозд А. В. Оценка контролепригодности цифровых компонентов встроенных систем критического применения /А. В. Дрозд В. С.Харченко, С. Г. Антошук, М. А. Дрозд, Ю. Ю Сулима. Радіоелектронні і комп'ютерні системи. – 2012. – №6. – С. 184-190.

16. ДСТУ ІЕС 60870-5-101:2014. Пристрої та системи телемеханіки. Частина 5-101. Протоколи передавання.

17. ДСТУ ІЕС 60870-5-104: 2014. Пристрої та системи телемеханіки. Частина 5-104. Протоколи передавання. Доступ до мережі згідно з ІЕС 60870-5-101 із використанням стандартних профілів передавання даних

18. ДСТУ ISO 9000:2007 (ISO 9000:2005, IDT) Системи управління якістю. Основні положення та словник термінів. Держспоживстандарт України, Київ, 2008. – 35 с.

19. Дунець Р.Б., Тиранський Д.Я. Дослідження часткової ре конфігурації ПЛІС // Радіоелектронні і комп'ютерні системи. – 2009. – №6(40). – С.240-244.

20. Дунець Р.Б., Тиранський Д.Я. Проблеми побудови частково реконфігурованих систем на ПЛІС // Радіоелектронні і комп'ютерні системи. – 2010. – №7(48). – С.200 – 204.

21. Жуков И.А., Дрововозов В.И., Способы повышения надежности и безопасности сбора информации в системах управления реального времени // Проблемы інформатизації та управління. –2008, 1(23). – С. 262– 276.

22. Зубинский А. Распыленная разумность // Компьютерное обозрение. – 2003. – № 8– С. 58-61.

23. Зюко А.Г. Помехоустойчивость и эффективность систем передачи информации. – М.: Радио и связь, 1985. – 272 с.

24. Казимир В. В. Проектування комп'ютерних систем на основі мікросхем програмованої логіки: монографія / С. А. Іванець, Ю. О. Зубань, В. В. Казимир, В. В. Литвинов. – Суми : Сумський державний університет, 2013. – 313 с.

25. Коды, исправляющие ошибки. Питерсон У., Уэлдон Э.: Мир, 1976. - 593 с.

26. Комплекс дистанційного керування технологічними процесами „СТРІЛА-М” на основі ПЕОМ Технічний опис та інструкція з експлуатації. Тернопіль, ТОВ «Стріла», 2014. – 54 с.

27. Компьютерные сети. Принципы, технологии, протоколы / Олифер В. Г., Олифер Н. А.– СПб.: Питер, 2007. – 960 с.

28. Кондратенко Ю.П., Мохор В.В., Сидоренко С.А. Verilog-HDL для моделирования и синтеза цифровых электронных схем. Учебное пособие. Под редакцией Ю.П. Кондратенко. – Николаев.: Издательство НГГУ им. Петра Могилы, 2002. – 221 с.

29. Краснобаев В. А. Метод обработки данных в классе вычетов / Краснобаев В. А., Янко А. С., Кошман С. А., Сомов С. А., Бендес Ю. П. // Збірник наукових праць Харківського університету Повітряних сил. – 2014. – №2 – С.121-126.

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

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

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

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

34. Міночкін А.І., Романюк В.А., Жук О.В. Перспективи розвитку тактичних сенсорних мереж// Збірник наукових праць №4. – К.: ВІПІ НТУУ “КПІ”. – 2007. – С. 112 – 119.

35. Муттер В.М. Основы помехоустойчивой телепередачи информации. -Л.: Энергоатомиздат. Ленингр. отд-ние, 1990. – 268 с.

36. Надёжность и эффективность в технике: Справочник в 10 т. Ред. совет: Авдеевский В.С. и др. Том 3. Эффективность технических систем. Под общ. ред. В.Ф.Уткина, Ю.В.Крючкова. – М.: Машиностроение, 1988. – 328 с.

37. Нейфах А.Э. Сверточные коды для передачи дискретной информации. – М.: Наука, 1979. – 222 с.

38. Николайчук Я. Н. Теорія джерел інформації – Тернопіль: ТЗОО «Тернограф» - 2010. – 534 с.

39. Николайчук Я.М. Алгоритм знаходження системи модулів модифікованої досконалої форми системи залишкових класів / Я.М.Николайчук, І.З. Якименко, М.М. Касянчук, Л.М.Тимошенко, Т.М.Долинюк // Матеріали Міжнародної науково-практичної конференції «Сучасні інформаційні та електронні технології». - Одеса, 26 – 30 травня 2014 р. – С. 115-116.

40. Николайчук Я.М. Проблемы реорганизации структуры процессоров у разных теоретико-числовых базисах / Я.М. Николайчук // Збірник матеріалів міжнародної наукової координаційної наради «Інформаційні проблеми комп'ютерних систем, юриспруденції, енергетики, економіки, моделювання та управління» (ICSM-2014). - Тернопіль, 2014.- С.110-114.

41. Николайчук Я.М. Теоретические основы, методы и процессоры преобразования информации в кодах поля Галуа на базе вертикально-информационной технологии / Я.М.Николайчук, П.В. Гуменный // Кибернетика и системный анализ. - Международный научно-теоретический журнал института кибернетики им. В.М. Глушкова НАН Украины. – 2014. – Том 50, №3. – С.17-26.

42. Николайчук Я.М. Теоретичні основи побудови та структура спецпроцесорів в базисі Крестенсона / Я.М. Николайчук, О.І Волинський, С.В.Кулина // Вісник Хмельницького національного університету. – 2007. - №3. – Т1. – С. 85-90.

43. Николайчук Я.Н. Теоретические основы аналитического вычисления коэффициентов базисных чисел преобразования Крестенсона/ Я.Н.Николайчук, І.З. Якименко, М.Н. Касянчук // Кибернетика и системный анализ. – 2014. - №5. – С. 3 – 8.

44. Николайчук Я.Н., Коды поля Галуа: теорія і використання /Монографія/ – Тернопіль: ТЗОО «Тернограф», -2012. –576 с.

45. Обобщенные каскадные помехоустойчивые конструкции на базе сверточных кодов. В.В. Зяблов, С.А. Шавгулидзе. - М.: Москва, 1991. – 207 с.

46. Опанасенко В.Н. Бортовые проблемно-ориентированные процессоры для аппаратной реализации алгоритмов управления космическими аппаратами / В.Н. Опанасенко, А.Н. Лисовый // Проблеми інформатизації та управління: Зб. наукових праць НАУ. – Вип. 3 (47). – Київ, 2014. – С. 70–74.

47. Опанасенко В.Н., Сахарин В.Г. Реконфигурируемые системы на современной элементной базе // Комп'ютерні засоби, мережі та системи. – Киев: Ін-т кибернетики ім. В.М. Глушкова НАН України, 2003. – №. 2. – С. 25–32.

48. Орищенко В.И. Санников. В.А., Свириденко В.А. Сжатие данных в системах сбора и передачи информации / Под ред. В.А. Свириденко. – М.: Радио и связь, 1985. –184 с.

49. Палагин А.В. Проектирование реконфигурируемых цифровых систем: монография / А.В. Палагин, А.А. Баркалов, В.Н. Опанасенко, Л.А. Титаренко. – Луганск: изд-во ВНУ им. В. Даля, 2011. – 432 с.

50. Палагин А.В., Опанасенко В.Н. О проектировании реконфигурируемых устройств на основе программируемых логических интегральных схем // Электронное моделирование. – 2006. – 28, № 4. – С. 65–74.

51. Палагин А.В., Опанасенко В.Н., Сахарин В.Г. Проектирование цифровых устройств на кристаллах ПЛИС с использованием Core – ядер // Вестн. СевГТУ. Автоматизация процессов и управление. – 2003. – Вып. 57. – С. 33–47.

52. Пристрій для перетворення паралельного двійкового коду в код системи залишкових класів. Патент №104912 України: МПК G06F 7/72 (2006.01), G06F 7/38 (2006.01), H03M 7/18 (2006.01) / Винахідники: Яцків В.В., Саченко А.О., Су Цзюнь / Власник: Тернопільський національний економічний університет – № а201204834; заявл.17.04.2012 р.; опубл. 25.03.2014р., Бюл.№ 6.

53. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. ван Стеен. – СПб.: Питер, 2003. – 877 с.

54. Романов В.О. Безпроводна сенсорна мережа для прецизійного землеробства та екологічного моніторингу / В.О.Романов, О.В. Палагін, І.Б.

Галелюка, О.В. Вороненко // Комп'ютерні засоби, мережі та системи. – 2014., –№13. – С.53-62.

55. Романюк В.А. Интеллектуальная маршрутизация в мобильных радиосетях (MANET) / Романюк В.А., Шацко П.В., Сова О.Я. [и др.] // Зв'язок. – 2011. – № 2(94). – С. 24 – 31.

56. Романюк В.А. Мобильные радиосети – перспективы беспроводных технологий // Сети и телекоммуникации. – 2003. – № 12. – С. 62 – 68.

57. Романюк В.А., Жук О.В., Сова О.Я. Аналіз протоколів маршрутизації в бездротових сенсорних мережах // Збірник наукових праць ВІТІ НТУУ „КПІ”. – 2008. – № 1. – С. 73 – 85.

58. Сетевое кодирование / Габидулин Э.М., Пилипчук Н.И., Колыбельников А.И. [та др.] // Труды МФТИ. – 2009. – Том 1, № 2 – С.3-28.

59. Синтез и анализ живучести сетевых систем: монографія / Ю.Ю. Громов, В.О. Драчев, К.А. Набатов, О.Г. Иванова. – М. : «Издательство Машиностроение-1», 2007. – 152 с.

60. Скляр Б. Цифровая связь. Теоретические основы и практическое применение, 2-е издание.: Пер. с англ. – М.: Издательський дом “Вильямс», 2003. – 1104 с.

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

62. Спосіб мережного кодування даних. Патент на корисну модель № 96835 України: МПК (2015.01), H0J 13/00, G06F 11/08 (2006.01) / Винахідник: Яцків В.В. / Власник: Яцків В.В. – № u2014 07525; заявл. 04.07.2014 р.; опубл. 25.02.2015 р., Бюл. № 4.

63. Столлингс В. Беспроводные линии связи и сети.: Пер. с англ. – М.: Издательський дом «Вильямс», 2003. – 640 с.

64. Су Цзюнь. Повышение эффективности передачи данных в беспроводных сенсорных сетях на основе многопутевой маршрутизации / Су Цзюнь, В.В. Яцкив, А.О.Саченко // Физика, математика, информатика. Вестник Брестского государственного технического университета, Брест.– 2010. – № 5 (650). – С. 21 – 24.

65. Су Цзюнь. Спецпроцессор кодирования изображений в системе остаточных классов / Су Цзюнь., В.В. Яцкив, А.А.Саченко, Ху Чежньбин // Труды МНПК «Современные информационные и электронные технологии», СИЭТ-2012. – Одесса, 2012. – С.95.

66. Су Цзюнь. Спосіб організація “зеленого коридору” для автомобілів спеціального призначення на основі ZigBee модулів/ Су Цзюнь, Яцків В.В. // Матеріали II Міжнародної науково-практичної конференції “Інформаційні технології та комп’ютерна інженерія” (26-27 травня 2011 р.). – Харків: ХНЕУ, 2011. – С.51-52.

67. Сэломон Д. Сжатие данных, изображений и звука. – М.: Техносфера. Серия: Мир программирования, 2004. – 368 с.

68. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. – СПб.: Питер, 2012. – 960 с.

69. Толстикова Е.В. Минимизация избыточности объема передачи данных в сети радиодатчиков // Проблемы інформатизації та управління. – 2010. – №1(29). – С.168-171.

70. Томас М. Томас П. Структура и реализация сетей на основе протокола OSPF, 2-е изд.: Пер. с англ. – М.: Издательский дом "Вильямс", 2004. – 816 с.

71. Тужилкин О.В., Ульянин Н.С. Методы оценки эффективности работы беспроводной сенсорной сети. Известия Южного федерального университета. Технические науки. – 2012. – № 130.5. – С.28-32.

72. Цаволюк Т.Г. Метод исправления ошибок на основе модулярных корректирующих кодов / Т.Г.Цаволюк, В.В.Яцкив // Физика, математика,

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

73. Цыбизов, А. А. Оценка эффективности сетей связи // Вестник Рязанского государственного радиотехнического университета. – 2009. – Вып. 3(29). – С. 19–24.

74. Чекмарев Ю. В. Вычислительные системы, сети и телекоммуникации. Издание второе, исправленное и дополненное. – М.: ДМК Пресс, 2009. – 184 с.

75. Червяков Н. И., Сахнюк П. А., Шапошников А. В., Ряднов С. А. Модулярные параллельные вычислительные структуры нейро- процессорных систем / Под. ред. Н.И. Червякова. – М.: ФИЗМАТЛИТ, 2003. 288 с.

76. Шевчук Б.М., Задірака В.К., Гнатів Л.О., Фраєр С.В. Технологія багатофункціональної обробки і передачі інформації в моніторингових мережах. – К.: Наук. думка, 2010. – 370 с.

77. Шевчук Б.М., Зінченко В.П., Фраєр С.В. Високоінформативна обробка та кодування сигналів і зображень в мережах дистанційного моніторингу станів об'єктів // Наукові вісті НТУУ «КПІ». – 2008. – № 2. – С. 89 – 96.

78. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Триумф, 2002. – 816 с.

79. Штовба Д. Муравьиные алгоритмы: теория и применение /Штовба Д. // Программирование. – 2005. – №4. – С.1-16.

80. Яцкив Н.Г. Метод кодирования данных в беспроводных компьютерных сетях на основе преобразования системы остаточных классов / Н.Г.Яцкив, В.В.Яцкив, Р.В. Крепич // Физика, математика, информатика. Вестник Брестского государственного технического университета – Брест, 2007. – № 5 (47). – С. 8–10.

81. Яцків В. В. Метод кодування зображень в системі залишкових класів / В. В. Яцків, Н. Г.Яцків // Труды МНПК «Современные

информационные и электронные технологии» (СИЭТ-2013). – Одесса, 2013. – С.44-46.

82. Яцків В. В. Методи виконання модулярних операцій та їх реалізація на ПЛІС / В. В. Яцків // Вісник Хмельницького національного університету. Технічні науки . - 2014. - № 6. - С. 218-224.

83. Яцків В. В. Проблеми створення комп'ютерних мереж на основі відкритих лазерних каналів зв'язку / В. В. Яцків // Тези міжнародної науково-технічної конференції. “Контроль і управління в складних системах”, КУСС - 2003. – Вінниця: «УНІВЕРСУМ – Вінниця». – 2003. – С. 70.

84. Яцків В.В. Алгоритм мережевого кодування даних на основі системи залишкових класів/ В.В.Яцків, Ю.С.Матіїшин, О. І. Крушельницький // АСІТ'2013, Тернопіль, 17-18 травня 2013. – С.84-85.

85. Яцків В.В. Безпроводні сенсорні мережі на основі оптичного каналу зв'язку // Науковий вісник Чернівецького університету: Фізика. Електроніка. Тематичний випуск “Комп'ютерні системи та компоненти” – Чернівці: ЧНУ.– 2008. – Вип. 426, Ч. II. – С. 153–156.

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

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

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

89. Яцків В.В. Завадозахищений метод передавання даних в безпроводних сенсорних мережах / В.В.Яцків, Н.Г.Яцків, Д.І.Боднар // Науковий вісник Чернівецького університету: Комп'ютерні системи та компоненти. – Чернівці: ЧНУ. – 2009. – Вип. 446. – С.117-120.

90. Яцків В.В. Контроль виконання арифметичних операцій на основі модулярних кодів /В.В.Яцків // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2015. – № 4(53). – С.135-138.

91. Яцків В.В. Концепція побудови безпроводних сенсорних мереж на основі колективного інтелекту / В.В.Яцків, Н.Г.Яцків // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2015. – № 2. – С.217-221.

92. Яцків В.В. Мережне кодування в базисі Крестенсона / Яцків В.В., Су Цзюнь, Дорош В.І. // Поступ в науку. Збірник наукових праць Бучацького інституту менеджменту і аудиту, Бучач. – 2011. – №7. – С.187 – 190.

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

94. Яцків В.В. Метод кодування та передавання мультимедійних даних в безпроводних сенсорних мережах/ В.В.Яцків, А.О. Саченко, Су Цзюнь // Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій: тези доповідей V Міжнародної науково-практичної конференції. – Запоріжжя: ЗНТУ, 2010. – С.133-135.

95. Яцків В.В. Метод мережного кодування в системі залишкових класів / В.В. Яцків // Науковий вісник "Комп'ютерні системи та мережі" Національного університету «Львівська політехніка» Львів. – 2013. - № 773. – С.157-164.

96. Яцків В.В. Метод передавання даних в сенсорних мережах / В.В.Яцків, Н.Г. Яцків // Труды десятой МНПК “Современные

информационные и электронные технологии”, СИЭТ-2009. – Одесса, 2009. – С.161.

97. Яцків В.В. Метод розподілу трафіку в безпроводних сенсорних мережах на основі системи залишкових класів /В.В.Яцків // Тези міжнародної науково - технічної конференції “Контроль і управління в складних системах”, КУСС – 2012, «УНІВЕРСУМ – Вінниця». – 2012. С. 39.

98. Яцків В.В. Метод стиснення зображень без втрат на основі залишків в безпроводних сенсорних мережах / В.В.Яцків, Н.Г. Яцків // Труды десятой МНПК “Современные информационные и электронные технологии” (СИЭТ-2010). – Одесса, 2010. – С.145.

99. Яцків В.В. Метод та пристрій кодування мультимедійних даних на основі системи залишкових класів/ В.В. Яцків // Праці Одеського політехнічного університету. Науковий та науково-виробничий збірник – 2013. – №1 (40). – С.61 - 66.

100. Яцків В.В. Методи кодування даних для фізичного рівня комп’ютерних мереж / В.В.Яцків, Ю.В. Кудряшов // Вісник національного університету “Львівська політехніка”. Радіоелектроніка та телекомунікації. – Львів. – 2002. – №443. – С. 131-134.

101. Яцків В.В. Методи кодування сигналів в безпроводних оптичних каналах зв’язку / В.В.Яцків, Н.Г.Яцків, Д.Я. Геник // Вісник Технологічного університету Поділля. - Хмельницький. – 2005. – № 4, Ч.1, Т. 1 – С. 246 – 248.

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

103. Яцків В.В. Напрямки підвищення ефективності роботи безпроводних сенсорних мереж / В.В. Яцків // Контроль і управління в складних системах (КУСС-2014). XII Міжнародна конференція. Тези доповідей. Вінниця, 14-16 жовтня 2014 року. – Вінниця: ВНТУ. – 2014. – С.77.

104. Яцків В.В. Принципи реалізації основних модулярних операцій на ПЛІС / В.В. Яцків // Матеріали VI – ої Українсько – польської науково-практичної конференції «Електроніка та інформаційні технології» ЕЛІТ-2014, Львів – Чинадієво, Україна. – С.20-22.

105. Яцків Н. Г. Методи кодування та фізичного представлення сигналів в інформаційно-керуючих системах/ Н.Г.Яцків, В. В. Яцків // Розвідка і розробка нафтових і газових родовищ. Серія: Методи і засоби технічної діагностики. – Івано-Франківськ: ІФДТУНГ. – 2001. – Т. 8, №38. – С. 140-144.

106. Яцків Н. Г. Спецпроцесор обробки даних на основі перетворення Крестенсона – Галуа / Н.Г. Яцків, Р.І. Король, В.В.Яцків, Т.Г.Федчишин/ Яцків Н. Г., Король Р. І., Яцків В. В., Федчишин Т. Г. // Вісник Технологічного університету Поділля. – 2003. –Т1, №3. – С. 105 – 108.

107. Яцків В.В. Метод підвищення надійності передачі даних в безпроводних сенсорних мережах на основі системи залишкових класів / Яцків В.В. // Радіоелектроніка та інформатика. – 2010. – №2. – С.32–35.

108. Abdelzaher T, He T, Stankovic J. Feedback control of data aggregation in sensor networks. In: In conference Decision and Control, 2004. CDC. 43rd IEEE Conference on. IEEE, 2004. – Pp. 1490-1495.

109. Abughalieh, N., Steenhaut, K., Nowé, A. Low power channel coding for wireless sensor networks. Communications and Vehicular Technology in the Benelux (SCVT), 2010 17th IEEE Symposium on. IEEE, 2010. – Pp. 1-5.

110. Agarwal R., Popovici E. M., O’Flynn B. Adaptive wireless sensor networks: A system design perspective to adaptive reliability // In Wireless Communications and Sensor Networks. – 2006. – Pp. 216-225.

111. Ahlswede R., Cai N., Li S.-Y.R., Yeung R.W. Network information flow //IEEE Trans. Inform. Theory. – 2000. – V. IT-46, N. 6. – Pp. 1204 – 1216.

112. Akyildiz F., Pompili D., Melodia T. Underwater Acoustic Sensor Networks: Research Challenges,” *Ad Hoc Net.*, vol. 3, no. 3, May 2005. – Pp. 257–79.
113. Akyildiz I. F., Melodia T., Chowdury K. R. Wireless multimedia sensor networks: applications and testbeds // *Proceedings of the IEEE* (invited paper), 2008. – Vol. 96. – № 10 – Pp. 1588-1605.
114. Akyildiz I. F., Vuran M. C. *Wireless Sensor Networks*, New York: John Wiley & Sons, 2010. – 571 p.
115. Akyildiz I.F Stuntebeck E.P. Wireless underground sensor networks: Research challenges, *Ad Hoc Networks Journal*, 2006. – №4.– Pp. 669–686.
116. Akyildiz I.F, Melodia T., Chowdhury K.R. A survey on wireless multimedia sensor networks. *Comput. Netw*, 2007.– №51 – Pp.921–960.
117. Akyildiz I.F., Su W., Sankarasubramaniam Y., Cayirci E. Wireless sensor networks: a survey. *Comput. Netw*, 2002. – №38. – Pp.393 – 422.
118. Akyildiz Ian F., Zhi Sun, Mehmet C. Vuran. Signal propagation techniques for wireless underground communication networks. *Physical Communication*, 2009. – №2. – Pp. 167–183.
119. Alrajeh N. et al. Error Correcting Codes in Wireless Sensor Networks: An Energy Perspective. *Applied Mathematics & Information Sciences*, 2015, 9.2. – Pp.809-818.
120. Al-suhail G. A., Louis K. W., Abdallah T. Y. Energy Efficiency Analysis of Adaptive Error Correction in Wireless Sensor Networks. *International Journal of Computer Science Issues*, Vol. 9, Issue 4, No 2, July 2012 – Pp. 79–84.
121. Ang Li-minn, et al. *Wireless Multimedia Sensor Networks on Reconfigurable Hardware*. Springer, 2013 – 283 p.
122. Antonio de la Piedra, An Braeken, Abdellah Touhafi. Sensor Systems Based on FPGAs and Their Applications: A Survey. *Sensors*, 2012, Vol.12 (9). – Pp.12235 – 12264.

123. Balakrishnan Gopinath, et al. Performance analysis of error control codes for wireless sensor networks. *Information Technology, 2007. ITNG'07. Fourth International Conference on. IEEE, 2007.* – Pp. 876–879.
124. Bao X., Li J. Matching Code-on-Graph with Network-on-Graph: Adaptive Network Coding for Wireless Relay Networks // *IEEE Transactions on Wireless Communications, 2008.* – V. 7, N. 2. – Pp. 574–583.
125. Bassoli, R., Marques, H., Rodriguez, J., Shum, K. W., & Tafazolli, R. Network coding theory: A survey. *Communications Surveys & Tutorials, IEEE, 2013.* – 15(4). – Pp.1950-1978.
126. Bin Qaisar S., Karande S., Misra, K., Radha, H. Optimally mapping an iterative channel decoding algorithm to a wireless sensor network. In *Communications, 2007. ICC'07. IEEE International Conference on IEEE 2007.* – Pp. 3283-3288.
127. Blum C. *Swarm intelligence in optimization / Blum C., Xiaodong Li // Springer Berlin Heidelberg.* – 2008. – Pp. 43-85.
128. Bykovyy P. A Low – Cost Network Controller for Security Systems /P. Bykovyy, I. Maykiv, I. Turchenko, O. Kochan, V. Yatskiv, G. Markowsky // *Proceeding of the International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: IDAACS'2005, 2005.* – Pp. 388 – 391.
129. Campobello G., Leonardi A., Palazzo S. Energy Saving and Reliability in Wireless Sensor Networks Using a CRT-based Packet Splitting Algorithm //University of Messina, Italy. – 2010.
130. Cayirci E. Data aggregation and dilution by modulus addressing in wireless sensor networks. *IEEE Communications Letters, 2003, 7.8.* – Pp.355-357.
131. Charbonneau Daniel, Anna Dornhaus. When doing nothing is something. How task allocation strategies compromise between flexibility, efficiency, and inactive agents. *Journal of Bioeconomics 17.3, 2015.* – Pp.217-242.

132. Chlamtac I., Petrioli C., Redi J. Energy-conserving go-back-N ARQ protocols for wireless data networks. In Proceedings of IEEE ICUPC'98, Piscataway, NJ, USA, October 1998, volume 2. – Pp. 1259–1263.

133. Chlamtac I., Petrioli C., Redi J. Energy-conserving selective repeat ARQ protocols for wireless data networks // Personal, Indoor and Mobile Radio Communications, 1998. The Ninth IEEE International Symposium on. – IEEE, 1998. – T. 2. – Pp. 836-840.

134. Chouhan, Sonali, Ranjan Bose, and M. Balakrishnan. Integrated energy analysis of error correcting codes and modulation for energy efficient wireless sensor nodes. *Wireless Communications, IEEE Transactions on* 8.10. 2009. – Pp.5348-5355.

135. Croce S., Marcelloni F., Vecchio M. Reducing power consumption in wireless sensor networks using a novel approach to data aggregation. *The Computer J.*, 2008. – Vol. 51, No. 2. – Pp. 227 – 239.

136. 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 – Pp. 481-497.

137. Dorigo M. Ant Colony Optimization /Dorigo M., T. Stützle // Cambridge, MA: MIT. Press/Bradford Books, 2004. – 321 p.

138. 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.–Pp. 181-188.

139. Eriksson O. Error Control in Wireless Sensor Networks: A Process Control Perspective. – 2011 / [Электронный ресурс] – Режим доступа: <http://www.diva-portal.org/smash/get/diva2:452988/fulltext01>

140. Fasolo E., Rossi M., Widmer J., Zorzi M. In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Trans. Wireless Commun.*, 2007. – Vol. 14, No. 2. – Pp. 70–87.

141. Fragouli C. Network coding: beyond throughput benefits, NetCod 2010, Proceedings of the IEEE, Volume 99, Issue 3, 2011, 28. – Pp. 461 – 475.
142. Fragouli C., Le Boudec J. Y., Widmer J. Network coding: an instant primer //ACM SIGCOMM Computer Communication Review. – 2006. – T. 36. – №. 1. – Pp. 63-68.
143. Fragouli C., Soljanin E. Information flow decomposition for network coding. Information Theory, IEEE Transactions on 52 (3), 2006. – Pp.829-848.
144. Fragouli C., Widmer J., J. Le Boudec. Efficient broadcasting using network coding. IEEE/ACM Transactions on Networking (TON), 2008. – 16 (2). – Pp.450-463.
145. Girod B., Aaron A., Rane S., Rebollo-Monedero D. Distributed video coding. Proc. IEEE 93(1), 2005. – Pp.71–83.
146. Goh, Vik Tor, Mohammad Umar Siddiqi. Multiple error detection and correction based on redundant residue number systems. Communications, *IEEE Transactions on* 56.3, 2008. – Pp.325-330.
147. Gray A., Lee C., Arabshahi P., Srinivasan J. Object-oriented reconfigurable processing for wireless networks. In: IEEE International Conference on Communications (ICC 2002), 2002, Vol. 1. – Pp. 497–501.
148. Guestrin C, Bodik P, Thibaux R, Paskin M, Madden S. Distributed regression: an efficient framework for modeling sensor network data. In: International conference on information processing in sensor networks (IPSN). Third International Symposium IEEE, 2004. – Pp. 1-10.
149. H. Qi, Y. Xu, X.Wang. Mobile-agent-based collaborative signal and information processing in sensor networks. Proceedings of the IEEE, 2003, Vol. 91, №8. – Pp. 1172–1183.
150. Hammoodi, I. S., Stewart, B. G., Kocian, A., & McMeekin, S. G. A comprehensive performance study of OPNET modeler for ZigBee wireless sensor networks. Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST'09. Third International Conference on. IEEE, 2009. – Pp. 357-362.

151. Howard S. L. Error control coding in low-power wireless sensor networks: When is ECC energy-efficient? / Howard, S. L., Schlegel C., Iniewski K. // EURASIP Journal on Wireless Communications and Networking, №2. – 2006. – Pp.29-29.

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

153. IEC 60870 -5-101: 2003 Telecontrol equipment and systems – Part 5: Transmission protocol, Section 101: Companion standard for basic telecontrol tasks.

154. IEC 60870-5-104: 2006 Telecontrol equipment and systems – Part 5: Transmission protocol, Section 104. Network access for IEC 60870-5-101 using standard transport profiles.

155. IEC Std. 60870-5-1: 2006 – Telecontrol equipment and systems. Part 5. Transmission protocols. Section 1. Transmission frame formats.

156. 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. – 203 p.

157. IEEE Standard for Part 15.4: Wireless Medium Access Control Layer (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs), IEEE Std 802.15.4- 2006.

158. IEEE Standard Verilog Hardware Description Language, IEEE Computer Society, IEEE, New York, NY, IEEE Std 1364 – 2001.

159. Jeong J., Cheng Tien Ee. Forward error correction in sensor networks. University of California at Berkeley, 2003. – Pp. 1-13.

160. Kan Yu, Gidlund M., Åkerberg J., Bjorkman M. Reliable and Low Latency Transmission in Industrial Wireless Sensor Networks. *Procedia Computer Science* 5, 2011. – Pp. 866–873.

161. Kang J., Zhou B., Ding Z., Lin S. LDPC coding schemes for error control in a multicast network // *IEEE International Symposium on Information Theory. Proc. ISIT-08.* – 2008. – Pp. 822 – 826.

162. Karvonen, H., Shelby, Z., Pomalaza-Raez C. Coding for energy efficient wireless embedded networks. In: *Wireless Ad-Hoc Networks, 2004 International Workshop on.* IEEE, 2004. – Pp. 300-304.

163. Kasirajan, Priya, Carl Larsen, and Sarangapani Jagannathan. A new data aggregation scheme via adaptive compression for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 2012. – Vol. 9, No. 1. – Pp. 5:1-5:26.

164. Kimura, N., Latifi, S. A survey on data compression in wireless sensor networks. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on IEEE, 2005.* - Vol. 2. – Pp. 8-13.

165. Kleinschmidt J. H., Borelli W. C., Pellenz M. E. An analytical model for energy efficiency of error control schemes in sensor networks. In *IEEE international conference on communications, Glasgow, Scotland, 2007.* – Pp. 3895–3900.

166. Kulkarni R. V., Förster A., Venayagamoorthy G. K. Computational intelligence in wireless sensor networks: a survey. *Communications Surveys & Tutorials, IEEE.* – 2011. – №13(1). – Pp.68-96.

167. Liao, Junsong, et al. FPGA based wireless sensor node with customizable event-driven architecture. *EURASIP Journal on Embedded Systems* 2013.1, 2013. – Pp.1-11.

168. Liu H. et al. Error control schemes for networks: An overview // *Mobile Networks and Applications.* – 1997. – T. 2. – № 2. – Pp. 167-182.

169. Lou W. An efficient N-to-1 mutlipath routing protocol in wireless

sensor networks. Proceedings of IEEE international Conference on Mobile Ad-hoc and Sensor Systems (MASS), Washington, DC, November 2005.

170. Lou W., Liu W., Fang Y. SPREAD: Enhancing data confidentiality in mobile ad hoc networks, IEEE INFOCOM 2004, HongKong, China, March, 2004.

171. M Chen, et al. Quality-driven secure audio transmissions in wireless multimedia sensor networks. Multimedia tools and applications, 2013, Vol. 67 (1). – Pp. 119-135.

172. Marcelloni F., Vecchio M. A simple algorithm for data compression in wireless sensor networks. Communications Letters, IEEE, 2008, 12.6. – Pp. 411-413.

173. Matsuda, T., Noguchi, T., & Takine, T. Survey of network coding and its applications. IEICE transactions on communications, 94(3), 2011. – Pp. 698-717.

174. Min Chen, et al. Mobile Agent-Based Directed Diffusion in Wireless Sensor Networks. EURASIP Journal on Advances in Signal Processing, 2007, Vol. 2007.1. – Pp.1-13.

175. Misra S., Reisslein M., Xue G. A survey of multimedia streaming in wireless sensor networks. IEEE Communications Surveys and Tutorials, Vol. 10, No. 1–4, 2008. – Pp. 18–39.

176. Mobile Ad Hoc Wireless Network for Pre- and Post-Emergency Situations in Nuclear Power Plant. Robert E. Hiromoto, A.Sachenko, V. Kochan, V.Koval, V. Turchenko, O. Roshchupkin, V. Yatskiv, K. Kovalok // The 2 nd IEEE International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems 11-12 September 2014, Offenburg, Germany. – P.92-96.

177. Molina J., Mora-Merchan J. M., Barbancho J., Leon C. Multimedia Data Processing and Delivery in Wireless Sensor Networks. Wireless Sensor Networks: Application-Centric Design, InTech, Chapter 23. – 2010. – Pp. 449–467.

178. Nguyen, D., Tran, T., Nguyen, T., & Bose, B. Wireless broadcast using network coding. *Vehicular Technology, IEEE Transactions on* 58(2), 2009. – Pp.914-925.

179. Nykolaychuk Ya. M. Theoretical Foundations for the Analytical Computation of Coefficients of Basic Numbers of Krestenson's Transformation / Nykolaychuk Ya. M., Kasianchuk, M. M., Yakymenko, I. Z. // *Cybernetics and Systems Analysis*, 2014, No. 50. – Pp.649-654.

180. Nykolaychuk, Y. M. Theory of Reliable and Secure Data Transmission in Sensory and Local Area Networks / Nykolaychuk, Y. M.; Shevchyuk, B. M.; Voronych, A. R.; Zavediuk, T. O.; Gladyuk, V. M. // *Cybernetics and Systems Analysis*, 2014, No. 50. – Pp. 304-315.

181. Nykolaychuk, Ya. M. Theoretical Bases, Methods, and Processors for Transforming Information in Galois Field Codes on the Basis of the Vertical Information Technology / Nykolaychuk, Ya. M., Humennij, P. V. // *Cybernetics and Systems Analysis*, 2014, No. 50. – Pp. 338-347.

182. Omondi A. Residue Number System: Theory and Implementation / A.Omond, B.Premkumar. Imperial College Press, 2007. – Vol. 2. – 296 p.

183. OPNET офіційний веб-сайт. – Режим доступу до сайту: <http://www.opnet.com>.

184. Pavlo Ilin, Jurgen Sieck, Volodymyr Brovko. Ad-hoc Media Façade. Proceedings of the 8-th 2015 IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS'2013, IDAACS'2013, Berlin, Germany, 2013, Volume 1. – Pp. 474-477.

185. Pham D.M., Aziz S. FPGA-based image processor architecture for wireless multimedia sensor network. In: 2011 IFIP 9th International Conference on Embedded and Ubiquitous Computing (EUC), 2011. – Pp. 100–105.

186. Pottie G. J., Kaiser W. J. Wireless integrated network sensors // *Communications of the ACM*. – 2000. – T. 43. – №. 5. – Pp. 51-58.

187. Radhakrishnan D., Y. Yuan A fast RNS Galois field multiplier. Circuits and Systems. IEEE International Symposium on IEEE. – 1990. – Pp. 2909-2912.

188. Razzaque M. A., Bleakley, C., Dobson, S. Compression in wireless sensor networks: A survey and comparative evaluation. ACM Transactions on Sensor Networks (TOSN), 2013. – Vol. 10, No. 1, Article 5. – Pp.1-44.

189. Roshanzadeh M., Saqaeeyan S. Error Detection & Correction in Wireless Sensor Networks By Using Residue Number Systems. International Journal of Computer Network and Information Security (IJCNIS) 4.2, 2012. – №2. – Pp. 29-35.

190. Sachenko A. Data Encoding in Residue Number System/ A. Sachenko, V. Yatskiv, R. Krepych, A. Karachka // Proceeding of the International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: IDAACS'2009, 2009. – Pp. 679 – 681.

191. Sachenko A. Modified Method of Noise-Immune Data Transmission in Wireless Sensors Networks / A.Sachenko, V. Yatskiv, R. Krepych // International Conference on Networks Security, Wireless Communications and Trusted Computing, “NSWCTC 2009”, Wuhan, China, 2009. – Vol. 2.– Pp.847-850.

192. Sankarasubramaniam Y. et al. Energy efficiency based packet size optimization in wireless sensor networks // Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. IEEE International Workshop on, 2003. – Pp. 1-8.

193. Sartipi, M., Fekri, F. Source and channel coding in wireless sensor networks using LDPC codes. Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON'2004. IEEE, 2004. – Pp. 309-316.

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

195. Shengli Z., Liew S. C., Lam, P. P. Physical layer network coding. arXiv preprint arXiv:0704.2475, 2007.
196. Sikora A., Groza V., Coexistence of IEEE 802.15.4 with other systems in the 2.4 ghz-ism-band, Instrumentation and Measurement Technology Conference, IMTC – 2005. Proceedings of the IEEE 3, 2005. – Pp. 1786 –1791.
197. Srisooksai T., Keamarungsi K., Lamsrichan P., Araki K. Practical data compression in wireless sensor networks: A survey. Journal of Network and Computer Applications, 2012. – Vol.35(1). – Pp.37-59.
198. Srivastava S., Spagnol C., Popovici E. Analysis of a set of error correcting schemes in multi-hop wireless sensor networks. In Research in Microelectronics and Electronics, 2009. PRIME 2009. – Pp. 1-4.
199. Stitt G., Vahid F., Nematbakhsh S. Energy savings and speedups from partitioning critical software loops to hardware in embedded systems. ACM Transactions on Embedded Computing Systems (TECS), 2004, Vol. 3 (1). – Pp. 218–232.
200. Su Jun. Data Transmission Optimal Routing in WSN Using Ant Colony Algorithm / Su Jun, Vasyl Yatskiv, Anatoly Sachenko, Nataliya Yatskiv // Proc. of the International Conf. TCSET 2012. – Lviv-Slavsko (Ukraine). – 2012. – Pp. 342– 343.
201. Su Jun. Improved Method of Ant Colonies to Search Independent Data Transmission Routes in WSN/ Su Jun, N. Yatskiv, A. Sachenko, V. Yatskiv // Proceedings 2012 IEEE 1st International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems. IDAACS-SWS-2012. – Offenburg, Germany. – 2012. – Pp. 52 – 57.
202. Su Jun. Method and Device for Image Coding & Transferring Based on Residue Number System / Su Jun, V.Yatskiv // Sensors & Transducers Journal, 2013. – Vol.18, Special Issue. – Pp.60-65.

203. Talooki, V. N., et al. Security concerns and countermeasures in network coding based communication systems: A survey. *Computer Networks*, 2015. – Vol. 83, No 4. – Pp. 422–445.

204. Tanenbaum E. *Raspredeleennyie systemy. Pryntsypy y paradyhmy* / E. Tanenbaum, M. van Steen. – SPb.: Pyter, 2003. – 877 p.

205. Tay Thian Fatt. A new algorithm for single residue digit error correction in Redundant Residue Number System / Tay Thian Fatt, Chang Chip-Hong // *Circuits and Systems (ISCAS), IEEE International Symposium IEEE*, 2014. – Pp. 1748-1751.

206. Tian Z., Yuan D.F., Liang Q.Q. Energy efficiency analysis of error control schemes in wireless sensor networks. In: *Wireless Communications and Mobile Computing Conference, 2008. IWCMC'08. International. IEEE, 2008.* – Pp. 401-405.

207. Todman T., Constantinides G., Wilton, S., Mencer, O., Luk, W., Cheung, P. Reconfigurable computing: Architectures and design methods. *IEE Proc. Computers and Digital Techniques*. 2005, Vol. 152 (2). – Pp.193 – 207.

208. Vuran M. C., Akyildiz I. F. Error control in wireless sensor networks: a cross layer analysis // *Networking, IEEE / ACM Transactions on.* – 2009. – T. 17. – №. 4. – Pp. 1186-1199.

209. Vuran M., Akyildiz I. Cross-layer analysis of error control in wireless sensor networks, *Sensor and Ad Hoc Communications and Networks*, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on 2, 2006. – Pp.585 –594.

210. Wang Y. Residue-to-Binary Converters Based On New Chinese Remainder Theorems. *IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing*, 2000 – Vol. 47, No. 3. – Pp.197–205.

211. Wang Y.C. Data compression techniques in wireless sensor networks. *Pervasive Computing*, 2012.

212. Yatskiv V. Concept of Designing the Wireless Sensor Networks on Ant Intelligence /V. Yatskiv, N. Yatskiv, A. Sachenko, O. Volynskyy // Proceedings of the 8-th 2015 IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS'2015, Warszawa, Poland, 2015, Volume 2. – Pp. 863-866.

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

214. Yatskiv V. Data Coding Method on the Basis of M – Sequences/ V. Yatskiv, N. Yatskiv // Proceeding of the International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: IDAACS '2009, 2009. – Pp. 698 – 700.

215. Yatskiv V. Data Transmission Method on the Basis of Spread Spectrum in Wireless Networks /V.Yatskiv, N Yatskiv// Proceedings of the VIIIth International Conference CADSM' 2005. – Lviv-Polyana, Ukraine. – 2005. – Pp. 166-167.

216. Yatskiv V. Improved Data Communication in WSN Using Modular Arithmetic /V. Yatskiv, A. Sachenko, N. Yatskiv // Wireless Communication and HTW - University of Applied Sciences Berlin. Berlin. – 2010. – Pp.39-49.

217. Yatskiv V. Multilevel Method of data coding in WSN /V. Yatskiv, Su Jun, N. Yatskiv, A. Sachenko, O. Osolinskiy // Proceeding of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications: IDAACS'2011, 2011. – Pp. 863 – 866.

218. Yatskiv V. Multiple Access on the Basis of Residue Number System Transformation / V. Yatskiv, N. Yatskiv// Proceeding of the International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: IDAACS'2005, 2005. – Pp. 527–530.

219. Yatskiv V. Multiple Error Detection and Correction Based on Modular Arithmetic Correcting Codes / V. Yatskiv, T. Tsavolyk, Hu Zhengbing // Proceedings of the 8-th 2015 IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS'2015, Warszawa, Poland, 2015, Volume 2. – Pp. 850-854.

220. Yatskiv V. Nonlinear data coding in wireless sensor networks/ V Yatskiv, Su Jun, N. Yatskiv, A. Sachenko // International Journal of Computing, 2011. – Vol. 10, Issue 4. – Pp. 383-390.

221. Yatskiv V. The network coding based on residue number system in wireless sensor networks /Vasyl Yatskiv, Su Jun // Proceedings of International Conference on Information Technology and Management Engineering ITME2011, September 23-25, 2011, Wuhan, China. – Pp.27-30.

222. Yatskiv V. The Network Coding Method Based on Residue Number System // Advanced Computer Systems and Networks: Design and Application. Proceedings of the 6-th International Conference. ACSN - 2013. – Lviv, Ukraine. – 2013. – Pp. 80 – 83.

223. Yatskiv V. The network coding method in wireless sensor networks based on residue number system. Zeszyty naukowe politechniki śląskiej. Seria: Organizacja i Zarządzanie z. 67. 2013. – Pp. 135-144.

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

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

226. Yatskiv V. Wireless Sensor Networks for Forest Fires Monitoring / V.Yatskiv, A. Sachenko // Computer Science and Information Technologies. Proceedings of IXth International Scientific and Technical Conference, CSIT 2014, Lviv, Ukraine, 18 – 22 November, 2014. – Pp.127.

227. Yeong Chee Mei, Naziri S. The FPGA Implementation of Multiplicative Inverse Value of $GF(2^8)$ Generator using Extended Euclid Algorithm (EEA) Method for Advanced Encryption Standard (AES) Algorithm. IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE), 2011. – Pp.12 – 15

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

229. Zhang X. Adaptive control and reconfiguration of mobile wireless sensor networks for dynamic multi-target tracking. IEEE Trans Autom Control, 2011. – №56(10). – Pp. 2429–2444.

230. Zhang Z. Network Coding Based on Chinese Remainder Theorem. arXiv preprint arXiv:1208.3966, 2012.

231. Zhong L. C., Rabaey J. M. An integrated data-link energy model for wireless sensor networks. In Proceedings of IEEE ICC'04, Paris, France, June 2004.

232. ZigBee Alliance. IEEE 802.15.4, ZigBee Standard. [Электронный ресурс] – Режим доступа: <http://www.zigbee.org>.

ДОДАТОК А

Таблиця А1 – Характеристики комерційних модулів безпроводних сенсорних мереж

Фірма	Назва модуля	Частота, ГГц	Струм спож. прийм. (Rx), мА	Струм спож. передач. (Tx), мА	ОЗП	ПЗП	Мікро-контролер	Інтер-фейси
Microchip	MRF24J40	2.4	19 (-94 dBm до +5 dBm)	23 (+0 dBm)	912 байт	–	Тільки трансівер	SPI
Cypress	CYRF7936	2,4	21 (-97 dBm до +4 dBm)	21 (+0 dBm)	–	–	Тільки трансівер	SPI
Texas Instruments	CC2650	2.4	5.9 (-97 dBm)	6.1 (+0 dBm), 9.1 (+5 dBm)	20 кБ SRAM + 8 кБ SRAM кеш	128 кБ	48 МГц ARM Cortex-M3	UART, I2C, I2S, 2xSPI, 12-bit ADC, 10/15/31 GPIO, Bluetooth Smart (BLE) і 6LoWPAN
Silicon Labs	EM358x	2.4	27 (-100 dBm)	31 (+3 dBm)	32-64 кБ	256-512 кБ	6/12/24 МГц Cortex-M3	USB2.0, UART, 2xSPI, 24 GPIO

Продовження таблиці Аі

Marvell	88MZ100	2.4	14 (-104 dBm)	26 (+9 dBm)	160 кБ	512 кБ	32/64 МГц Cortex-M3	2xUART, 2xSPI, 2xI2C, 31 GPIO
Freescale	MC1323x	2.4	34 (-94 dBm)	27 (0 dBm)	8 кБ	128 кБ	32 МГц HCS08QE	UART, SPI, I2C, 32 GPIO
NXP	JN-516x	2.4	17 (-95 dBm)	15 (+2.5 dBm)	8-32 кБ	64 - 512 кБ	32 МГц, 32-х бітний RISC проц.	2xUART, SPI, 2xI2C, 20 GPIO
Atmel	ATZB- 24-B0	2.4	21.8 (-101 dBm)	20.8 (0 dBm)	8 кБ	128 кБ	4 МГц ATmega128 1V	USB2.0, UART, USART, I2C, SPI, 30 GPIO
Telink	TLSR 8636	2.4	12 (-99 dBm)	12 (0 dBm)	16 кБ	512 кБ	48 МГц 32bit MCU	USB2.0, UART, I2C, SPI, 35/20 GPIO

ДОДАТОК Б

Текст програми пошуку коефіцієнтів

```
// boom.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include <iostream>
#include <conio.h>

using namespace std;

int main()
{
    int p,kilPar=0;
    cout<<"Vyvodyti vsi v1, v2,v3 i otrymani znachenia K"<<endl<<"Vvedit P = ";
    cin>>p;
    for (int v1=1; v1<p; v1++)
    for(int v2=1; v2<p; v2++)
    for(int v3=1;v3<p;v3++)
    {
        if(!(v1==v2 || v2==v3 || v3==v1))
        {
            //      cout<<endl<<endl<<"v1 = "<<v1<<" v2 = "<<v2<<" v3 = "<<v3<<endl;/
            bool a = true;
            int mas[42];
            int n =0;
            for (int e=1; e<=7; e++)
            {
                if (a)
                {
                    int K[6];
                    K[0]=(v1*e)%p;
                    K[1]=(v1*e*-1)%p;
                    K[2]=(v2*e)%p;
                    K[3]=(v2*e*-1)%p;
                    K[4]=(v3*e)%p;
                    K[5]=(v3*e*-1)%p;
                    for (int i=0;i<6;i++)
                    {
                        if(K[i]<0) K[i]=p-(K[i]*-1);
                        //cout<<K[i]<<" ";
                        for (int j = 0; j<n; j++)
                            if (mas[j]==K[i]) { a= false; break;}
                    }
                    if (a)
                    {
```

```
                mas[n] = K[i]; n++;
            }
            else break;
        }
        } else break;
    }
    if (a) { cout<<endl<<"++++"<<v1<<" "<<v2<<" "<<v3<<endl; kilPar++; }
//getch(); }
    }
}
cout<<endl<<"Kilkist znachen v1, v2, v3 - "<<kilPar;
getch();
return 0;
}
```

ДОДАТОК В

Таблиця Б.1 – Значення синдромів

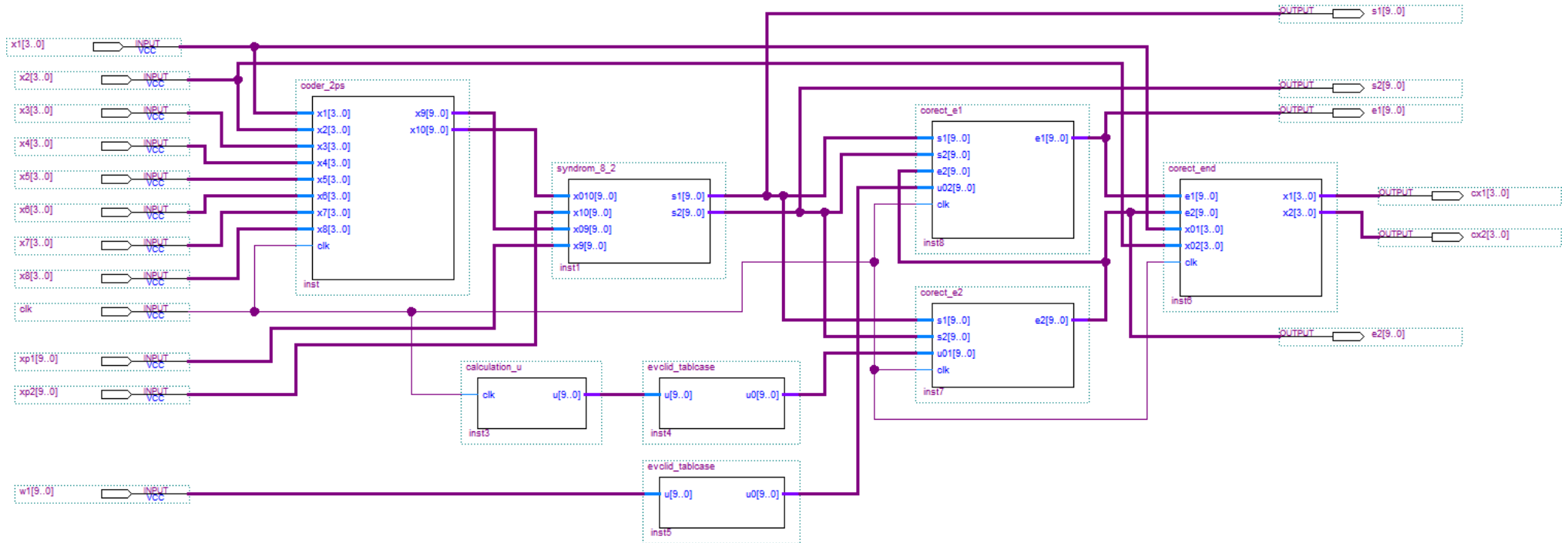
A1=V1*b1(modP)			A2=V2*b2(modP)			A3=V3*b3(modP)			A4=V4*b4(modP)			A5=V5*b5(modP)			A6=V6*b6(modP)			A7=V7*b7(modP)			A8=V8*b8(modP)		
V1	b1	A1	V2	b2	A2	V3	b3	A3	V4	b4	A4	V5	b5	A5	V6	b6	A6	V7	b7	A7	V8	b8	A8
13	0	0	17	0	0	19	0	0	23	0	0	29	0	0	31	0	0	37	0	0	43	0	0
	1	13		1	17		1	19		1	23		1	29		1	31		1	37		1	43
	2	26		2	34		2	38		2	46		2	58		2	62		2	74		2	86
	3	39		3	51		3	57		3	69		3	87		3	93		3	111		3	129
	4	52		4	68		4	76		4	92		4	116		4	124		4	148		4	172
	5	65		5	85		5	95		5	115		5	145		5	155		5	185		5	215
	6	78		6	102		6	114		6	138		6	174		6	186		6	222		6	258
	7	91		7	119		7	133		7	161		7	203		7	217		7	259		7	301
	8	104		8	136		8	152		8	184		8	232		8	248		8	296		8	344
	9	117		9	153		9	171		9	207		9	261		9	279		9	333		9	387
	10	130		10	170		10	190		10	230		10	290		10	310		10	370		10	430
	11	143		11	187		11	209		11	253		11	319		11	341		11	407		11	473
	12	156		12	204		12	228		12	276		12	348		12	372		12	444		12	516
	13	169		13	221		13	247		13	299		13	377		13	403		13	481		13	559
	14	182		14	238		14	266		14	322		14	406		14	434		14	518		14	602
	15	195		15	255		15	285		15	345		15	435		15	465		15	555		15	645
	-1	1008		-1	1004		-1	1002		-1	998		-1	992		-1	990		-1	984		-1	978
	-2	995		-2	987		-2	983		-2	975		-2	963		-2	959		-2	947		-2	935
	-3	982		-3	970		-3	964		-3	952		-3	934		-3	928		-3	910		-3	892
	-4	969		-4	953		-4	945		-4	929		-4	905		-4	897		-4	873		-4	849
	-5	956		-5	936		-5	926		-5	906		-5	876		-5	866		-5	836		-5	806
	-6	943		-6	919		-6	907		-6	883		-6	847		-6	835		-6	799		-6	763
	-7	930		-7	902		-7	888		-7	860		-7	818		-7	804		-7	762		-7	720
	-8	917		-8	885		-8	869		-8	837		-8	789		-8	773		-8	725		-8	677
	-9	904		-9	868		-9	850		-9	814		-9	760		-9	742		-9	688		-9	634

Продовження таблиці В.1

	-10	891		-10	851		-10	831		-10	791		-10	731		-10	711		-10	651		-10	591
	-11	878		-11	834		-11	812		-11	768		-11	702		-11	680		-11	614		-11	548
	-12	865		-12	817		-12	793		-12	745		-12	673		-12	649		-12	577		-12	505
	-13	852		-13	800		-13	774		-13	722		-13	644		-13	618		-13	540		-13	462
	-14	839		-14	783		-14	755		-14	699		-14	615		-14	587		-14	503		-14	419
	-15	826		-15	766		-15	736		-15	676		-15	586		-15	556		-15	466		-15	376

ДОДАТОК Г

Функціональна схема декодера



ДОДАТОК Д

Verilog-код декодера модулярного коректуючого коду

```

module decoder_8_4 (x01,x02,x03,x04,x05,x06,x07,x08,x1,x2,x3,x4,x5,x6,x7,x8,s);
    input [3:0] x01,x02,x03,x04,x05,x06,x07,x08;
    input wire [9:0] s;
    output wire [3:0]x1,x2,x3,x4,x5,x6,x7,x8;
    error_1(x1,x01,s);
    error_2(x2,x02,s);
    error_3(x3,x03,s);
    error_4(x4,x04,s);
    error_5(x5,x05,s);
    error_6(x6,x06,s);
    error_7(x7,x07,s);
    error_8(x8,x08,s);
endmodule

module error_1(x1,x01,s);
    output reg[3:0] x1;
    input [3:0] x01;
    input [9:0] s;

    always @(*)
        begin
            if (s==13)
                begin
                    x1=(x01-1);
                end
            else if (s==26)
                begin
                    x1=(x01 -2);
                end
            else if (s==39)
                begin
                    x1=(x01-3);
                end
            else if (s==52)
                begin
                    x1=(x01-4);
                end
            else if (s==65)
                begin
                    x1=(x01-5);
                end
        end
endmodule

```



```
else if (s==78)
begin
x1=(x01-6);
end
else if (s==91)
begin
x1=(x01-7);
end
else if (s==104)
begin
x1=(x01-8);
end

else if (s==117)
begin
x1=(x01 -9);
end
else if (s==130)
begin
x1=(x01-10);
end
else if (s==143)
begin
x1=(x01-11);
end
else if (s==156)
begin
x1=(x01-12);
end
else if (s==169)
begin
x1=(x01-13);
end
else if (s==182)
begin
x1=(x01-14);
end
else if (s==195)
begin
x1=(x01-15);
end
else if (s==1008)
begin
x1=(x01+1);
end
```

```
else if (s==995)
begin
x1=(x01+2);
end
else if (s==982)
begin
x1=(x01+3);
end
else if (s==969)
begin
x1=(x01+4);
end
else if (s==956)
begin
x1=(x01+5);
end
else if (s==943)
begin
x1=(x01+6);
end
else if (s==930)
begin
x1=(x01+7);
end
else if (s==917)
begin
x1=(x01+8);
end
else if (s==904)
begin
x1=(x01+9);
end
else if (s==891)
begin
x1=(x01+10);
end
else if (s==878)
begin
x1=(x01+11);
end
else if (s==865)
begin
x1=(x01+12);
end
else if (s==852)
```

```
begin
  x1=(x01+13);
end
else if (s==839)
begin
  x1=(x01+14);
end
else if (s==826)
begin
  x1=(x01+15);
end
else
begin
  x1= x01 [3:0];
end
end
endmodule
```

```
module error_2(x2,x02,s);
  output reg[3:0] x2;
  input [3:0] x02;
  input [9:0] s;

  always @(*)
  begin
    if (s==17)
    begin
      x2=(x02-1);
    end
    else if (s==34)
    begin
      x2=(x02 -2);
    end
    else if (s==51)
    begin
      x2=(x02-3);
    end
    else if (s==68)
    begin
      x2=(x02-4);
    end
    else if (s==85)
    begin
      x2=(x02-5);
    end
  end
end
```

```
else if (s==102)
begin
x2=(x02-6);
end
else if (s==119)
begin
x2=(x02-7);
end
else if (s==136)
begin
x2=(x02-8);
end

else if (s==153)
begin
x2=(x02 -9);
end
else if (s==170)
begin
x2=(x02-10);
end
else if (s==187)
begin
x2=(x02-11);
end
else if (s==204)
begin
x2=(x02-12);
end
else if (s==221)
begin
x2=(x02-13);
end
else if (s==238)
begin
x2=(x02-14);
end
else if (s==255)
begin
x2=(x02-15);
end
else if (s==1004)
begin
x2=(x02+1);
end
```

```
else if (s==987)
begin
x2=(x02+2);
end
else if (s==970)
begin
x2=(x02+3);
end
else if (s==953)
begin
x2=(x02+4);
end
else if (s==936)
begin
x2=(x02+5);
end
else if (s==919)
begin
x2=(x02+6);
end
else if (s==902)
begin
x2=(x02+7);
end

else if (s==885)
begin
x2=(x02+8);
end
else if (s==868)
begin
x2=(x02+9);
end
else if (s==851)
begin
x2=(x02+10);
end
else if (s==834)
begin
x2=(x02+11);
end
else if (s==817)
begin
x2=(x02+12);
end
```

```

        else if (s==800)
        begin
        x2=(x02+13);
        end
        else if (s==783)
        begin
        x2=(x02+14);
        end
        else if (s==766)
        begin
        x2=(x02+15);
        end

        else
        begin
        x2= x02 [3:0];
        end
end
endmodule

module error_3 (x3,x03,s);
    output reg[3:0] x3;
    input [3:0] x03;
    input [9:0] s;
    // reg [2:0] d3; // Оголошення 3 - розрядного регістра

    always @(*)
    begin
        if (s==19)
        begin
        x3=(x03-1);
        end
        else if (s==38)
        begin
        x3=(x03 -2);
        end
        else if (s==57)
        begin
        x3=(x03-3);
        end
        else if (s==76)
        begin
        x3=(x03-4);
        end
        else if (s==95)

```

```
begin
x3=(x03-5);
end
else if (s==114)
begin
x3=(x03-6);
end
else if (s==133)
begin
x3=(x03-7);
end
else if (s==152)
begin
x3=(x03-8);
end

else if (s==171)
begin
x3=(x03 -9);
end
else if (s==190)
begin
x3=(x03-10);
end
else if (s==209)
begin
x3=(x03-11);
end
else if (s==228)
begin
x3=(x03-12);
end
else if (s==247)
begin
x3=(x03-13);
end
else if (s==266)
begin
x3=(x03-14);
end
else if (s==285)
begin
x3=(x03-15);
end
else if (s==1002)
```

```
begin
x3=(x03+1);
end
else if (s==983)
begin
x3=(x03+2);
end
else if (s==964)
begin
x3=(x03+3);
end
else if (s==945)
begin
x3=(x03+4);
end
else if (s==926)
begin
x3=(x03+5);
end
else if (s==907)
begin
x3=(x03+6);
end
else if (s==888)
begin
x3=(x03+7);
end
else if (s==869)
begin
x3=(x03+8);
end
else if (s==850)
begin
x3=(x03+9);
end
else if (s==831)
begin
x3=(x03+10);
end
else if (s==812)
begin
x3=(x03+11);
end
else if (s==793)
begin
```



```

        x3=(x03+12);
    end
    else if (s==774)
    begin
        x3=(x03+13);
    end
    else if (s==755)
    begin
        x3=(x03+14);
    end
    else if (s==736)
    begin
        x3=(x03+15);
    end
    else
    begin
        x3= x03 [3:0];
    end

end
endmodule

module error_4 (x4,x04,s);
    output reg[3:0] x4;
    input [3:0] x04;
    input [9:0] s;
    // reg [2:0] d3; // Оголошення 3 - розрядного регістра

    always @(*)
    begin
        if (s==23)
        begin
            x4=(x04-1);
        end
        else if (s==46)
        begin
            x4=(x04 -2);
        end
        else if (s==69)
        begin
            x4=(x04-3);
        end
        else if (s==92)
        begin
            x4=(x04-4);
        end
    end
endmodule

```

```
end
else if (s==115)
begin
x4=(x04-5);
end
else if (s==138)
begin
x4=(x04-6);
end
else if (s==161)
begin
x4=(x04-7);
end
else if (s==184)
begin
x4=(x04-8);
end
else if (s==207)
begin
x4=(x04 -9);
end
else if (s==230)
begin
x4=(x04-10);
end
else if (s==253)
begin
x4=(x04-11);
end
else if (s==276)
begin
x4=(x04-12);
end
else if (s==299)
begin
x4=(x04-13);
end
else if (s==322)
begin
x4=(x04-14);
end
else if (s==345)
begin
x4=(x04-15);
end
```

```
else if (s==998)
begin
x4=(x04+1);
end
else if (s==975)
begin
x4=(x04+2);
end
else if (s==952)
begin
x4=(x04+3);
end
else if (s==929)
begin
x4=(x04+4);
end
else if (s==906)
begin
x4=(x04+5);
end
else if (s==883)
begin
x4=(x04+6);
end
else if (s==860)
begin
x4=(x04+7);
end
else if (s==837)
begin
x4=(x04+8);
end
else if (s==814)
begin
x4=(x04+9);
end
else if (s==791)
begin
x4=(x04+10);
end
else if (s==768)
begin
x4=(x04+11);
end
else if (s==745)
```

```

begin
x4=(x04+12);
end
else if (s==722)
begin
x4=(x04+13);
end
else if (s==699)
begin
x4=(x04+14);
end
else if (s==676)
begin
x4=(x04+15);
end
else
begin
x4= x04 [3:0];
end

end
endmodule

module error_5 (x5,x05,s);
output reg[3:0] x5;
input [3:0] x05;
input [9:0] s;
// reg [2:0] d3; // Оголошення 3 - розрядного регістра

always @(*)
begin
if (s==29)
begin
x5=(x05-1);
end
else if (s==58)
begin
x5=(x05 -2);
end
else if (s==87)
begin
x5=(x05-3);
end
else if (s==116)
begin

```

```
x5=(x05-4);
end
else if (s==145)
begin
x5=(x05-5);
end
else if (s==174)
begin
x5=(x05-6);
end
else if (s==203)
begin
x5=(x05-7);
end
else if (s==232)
begin
x5=(x05-8);
end

else if (s==261)
begin
x5=(x05 -9);
end
else if (s==290)
begin
x5=(x05-10);
end
else if (s==319)
begin
x5=(x05-11);
end
else if (s==348)
begin
x5=(x05-12);
end
else if (s==377)
begin
x5=(x05-13);
end
else if (s==406)
begin
x5=(x05-14);
end
else if (s==435)
begin
```

```
x5=(x05-15);
end
else if (s==992)
begin
x5=(x05+1);
end
else if (s==963)
begin
x5=(x05+2);
end
else if (s==934)
begin
x5=(x05+3);
end
else if (s==905)
begin
x5=(x05+4);
end
else if (s==876)
begin
x5=(x05+5);
end
else if (s==847)
begin
x5=(x05+6);
end
else if (s==818)
begin
x5=(x05+7);
end

else if (s==789)
begin
x5=(x05+8);
end
else if (s==760)
begin
x5=(x05+9);
end
else if (s==731)
begin
x5=(x05+10);
end
else if (s==702)
begin
```

```

        x5=(x05+11);
    end
    else if (s==673)
    begin
        x5=(x05+12);
    end
    else if (s==644)
    begin
        x5=(x05+13);
    end
    else if (s==615)
    begin
        x5=(x05+14);
    end
    else if (s==586)
    begin
        x5=(x05+15);
    end

    else
    begin
        x5= x05 [3:0];
    end

end
endmodule

```

```

module error_6 (x6,x06,s);

```

```

    output reg[3:0] x6;
    input [3:0] x06;
    input [9:0] s;

```

```

always @(*)
begin
    if (s==31)
    begin
        x6=(x06-1);
    end
    else if (s==62)
    begin
        x6=(x06 -2);
    end
    else if (s==93)
    begin

```

```
x6=(x06-3);
end
else if (s==124)
begin
x6=(x06-4);
end
else if (s==155)
begin
x6=(x06-5);
end
else if (s==186)
begin
x6=(x06-6);
end
else if (s==217)
begin
x6=(x06-7);
end
else if (s==248)
begin
x6=(x06-8);
end

else if (s==279)
begin
x6=(x06 -9);
end
else if (s==310)
begin
x6=(x06-10);
end
else if (s==341)
begin
x6=(x06-11);
end
else if (s==372)
begin
x6=(x06-12);
end
else if (s==403)
begin
x6=(x06-13);
end
else if (s==434)
begin
```



```
x6=(x06-14);
end
else if (s==465)
begin
x6=(x06-15);
end
else if (s==990)
begin
x6=(x06+1);
end
else if (s==959)
begin
x6=(x06+2);
end
else if (s==928)
begin
x6=(x06+3);
end
else if (s==897)
begin
x6=(x06+4);
end
else if (s==866)
begin
x6=(x06+5);
end
else if (s==835)
begin
x6=(x06+6);
end
else if (s==804)
begin
x6=(x06+7);
end

else if (s==773)
begin
x6=(x06+8);
end
else if (s==742)
begin
x6=(x06+9);
end
else if (s==711)
begin
```

```
        x6=(x06+10);
    end
    else if (s==680)
    begin
        x6=(x06+11);
    end
    else if (s==649)
    begin
        x6=(x06+12);
    end
    else if (s==618)
    begin
        x6=(x06+13);
    end
    else if (s==587)
    begin
        x6=(x06+14);
    end
    else if (s==556)
    begin
        x6=(x06+15);
    end

    else
    begin
        x6= x06 [3:0];
    end

end
endmodule
```

```
module error_7 (x7,x07,s);

    output reg[3:0] x7;
    input [3:0] x07;
    input [9:0] s;

    always @(*)
    begin
        if (s==37)
        begin
            x7=(x07-1);
        end
        else if (s==74)
```

```
begin
x7=(x07 -2);
end
else if (s==111)
begin
x7=(x07-3);
end
else if (s==148)
begin
x7=(x07-4);
end
else if (s==185)
begin
x7=(x07-5);
end
else if (s==222)
begin
x7=(x07-6);
end
else if (s==259)
begin
x7=(x07-7);
end
else if (s==296)
begin
x7=(x07-8);
end

else if (s==333)
begin
x7=(x07 -9);
end
else if (s==370)
begin
x7=(x07-10);
end
else if (s==407)
begin
x7=(x07-11);
end
else if (s==444)
begin
x7=(x07-12);
end
else if (s==481)
```

```
begin
x7=(x07-13);
end
else if (s==518)
begin
x7=(x07-14);
end
else if (s==555)
begin
x7=(x07-15);
end
else if (s==984)
begin
x7=(x07+1);
end
else if (s==947)
begin
x7=(x07+2);
end
else if (s==910)
begin
x7=(x07+3);
end
else if (s==873)
begin
x7=(x07+4);
end
else if (s==836)
begin
x7=(x07+5);
end
else if (s==799)
begin
x7=(x07+6);
end
else if (s==762)
begin
x7=(x07+7);
end

else if (s==725)
begin
x7=(x07+8);
end
else if (s==688)
```

```
begin
  x7=(x07+9);
end
else if (s==651)
begin
  x7=(x07+10);
end
else if (s==614)
begin
  x7=(x07+11);
end
else if (s==577)
begin
  x7=(x07+12);
end
else if (s==540)
begin
  x7=(x07+13);
end
else if (s==503)
begin
  x7=(x07+14);
end
else if (s==466)
begin
  x7=(x07+15);
end

else
begin
  x7= x07 [3:0];
end
end
endmodule
```

```
module error_8 (x8,x08,s);
```

```
output reg[3:0] x8;
input [3:0] x08;
input [9:0] s;
```

```
always @(*)
```

```
begin
  if (s==43)
begin
```

```
x8=(x08-1);
end
else if (s==86)
begin
x8=(x08 -2);
end
else if (s==129)
begin
x8=(x08-3);
end
else if (s==172)
begin
x8=(x08-4);
end
else if (s==215)
begin
x8=(x08-5);
end
else if (s==258)
begin
x8=(x08-6);
end
else if (s==301)
begin
x8=(x08-7);
end
else if (s==344)
begin
x8=(x08-8);
end

else if (s==387)
begin
x8=(x08 -9);
end
else if (s==430)
begin
x8=(x08-10);
end
else if (s==473)
begin
x8=(x08-11);
end
else if (s==516)
begin
```

```
x8=(x08-12);
end
else if (s==559)
begin
x8=(x08-13);
end
else if (s==602)
begin
x8=(x08-14);
end
else if (s==645)
begin
x8=(x08-15);
end
else if (s==978)
begin
x8=(x08+1);
end
else if (s==935)
begin
x8=(x08+2);
end
else if (s==892)
begin
x8=(x08+3);
end
else if (s==849)
begin
x8=(x08+4);
end
else if (s==806)
begin
x8=(x08+5);
end
else if (s==763)
begin
x8=(x08+6);
end
else if (s==720)
begin
x8=(x08+7);
end

else if (s==677)
begin
```

```
x8=(x08+8);
end
else if (s==634)
begin
x8=(x08+9);
end
else if (s==591)
begin
x8=(x08+10);
end
else if (s==548)
begin
x8=(x08+11);
end
else if (s==505)
begin
x8=(x08+12);
end
else if (s==462)
begin
x8=(x08+13);
end
else if (s==419)
begin
x8=(x08+14);
end
else if (s==376)
begin
x8=(x08+15);
end

else
begin
x8= x08 [3:0];
end

end
endmodule
```


ДОДАТОК Е

Програмний код генератора Verilog-коду

Веб інтерфейс генератора (код HTML)

```

<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width">
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>Generator</title>
  <link rel="stylesheet" type="text/css" href="style.css">
  <script type="text/javascript" src="jquery/jquery-2.1.4.min.js"></script>
  <script type="text/javascript" src="nicescroll/jquery.nicescroll.js"></script>
  <script type="text/javascript" src="script.js"></script>
</head>
<body>
  <div class="bg-container">
    <div class="header">
      <div class="container">
        </div>
      </div>
      <div class="main">
        <div class="container">
          <div class="params">
            <h2>Генератор коду на мові V-HDL</h2>
            <form class="parform" name="parform">
              <div class="form-group">
                <input type="text" id="quantity" name="quantity" value=""
placeholder="Кількість символів">
              </div>
              <div class="form-group">
                <input type="text" id="bit" name="bit" value="" placeholder="Розрядність
символів">
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

    <div class="form-group">
      <input type="text" id="coefficient" name="coefficient" value=""
placeholder="Коефіцієнти(наприклад: 3,5,7...)">
    </div>
    <div class="form-group">
      <input type="text" id="module" name="module" value="" placeholder="Модуль">
    </div>
    <div class="form-group">
      <input class="btn" type="button" id="submit" value="Submit">
      <input class="btn" type="reset" id="reset" value="Reset">
    </div>
  </form>
  <div class="clearfix"></div>
</div>
</div>
</div>
</div>
<div class="output-con" id="output-con">
  <div class="container">
  </div>
</div>
</body>
</html>

```

Графічний інтерфейс генератора (код CSS)

```

* {
  font-family: cursive;
}
html, body {
  margin: 0;
  padding: 0;
  width: 100%;
  height: 100%;
  /*margin-bottom: 50px;*/

```

```
        color: #ccc;
    }
.clearfix {
    content: "";
    display: table;
    clear: both;
}
.bg-container {
    background: url("Math-slider-image.jpg") no-repeat;
    background-size: cover;
    height: 100%;
    width: 100%;
}
.container {
    width: 1170px;
    margin: 0 auto;
    min-width: 320px;
}
.header {
    height: 1px;
    width: 100%;
    background-color: rgba(15,18,25,0);
}
.params {
    width: 585px;
    margin: 10% auto 0 auto;
    background-color: rgb(224, 224, 224);
    border-radius: 2px;
    box-shadow: 0px 0px 33px 0px #27272F;
}
h2 {
    margin: 0;
    padding: 20px 0;
    text-align: center;
```

```
    color: #3A3A46;
    margin-bottom: 15px;
}
.params form {
    margin: 0 auto;
    width: 70%;
}
::-webkit-input-placeholder {color:#84848E;}
::-moz-placeholder      {color:#84848E;}/* Firefox 19+ */
:-moz-placeholder      {color:#84848E;}/* Firefox 18- */
:-ms-input-placeholder  {color:#84848E;}
input[type="text"] {
    width: 95%;
    height: 35px;
    border: 0;
    border-radius: 2px;
    background-color: #E0E0E0;
    box-shadow: 0px 0px 1px 1px #ADADB1;
    padding: 2px 10px;
    color: #3A3A46;
}
input[type="text"]:focus {
    outline-color: #3A3A46;
}
.params .form-group {
    text-align: center;
    padding-bottom: 20px;
}
.btn {
    width: 180px;
    height: 40px;
    font-size: 17px;
    border-radius: 2px;
    border: 0;
```

```
        color: #fff;
        cursor: pointer;
        outline: 0;
        margin-top: 15px;
    }
input[type="button"] {
    margin-right: 20px;
    background-color: #C37D36;
}
input[type="button"]:active {
    border: 1px solid #7F4B16;
}
input[type="reset"] {
    background-color: #3E5486;
}
input[type="reset"]:active {
    border: 1px solid #213050;
}
.output-con {
    display: none;
    width: 100%;
    height: auto;
    background-color: #D1D1D1;
    padding: 40px 0;
}
.output {
    background-color: #E0E0E0;
    border-radius: 2px;
    padding: 20px;
    margin-bottom: 40px;
    overflow: auto;
    box-shadow: 0px 0px 15px 0px #27272F;
    color: #3A3A46;
}
```

```

.output:last-child {
    margin-bottom: 0;
}
.output pre {
    font-size: 12px;
}
@media screen and (max-width: 1169px) {
    .container {
        width: 100%;
    }
    .output {
        margin: 0 10px 40px;
    }
}
@media screen and (max-width: 600px) {
    .params {
        width: 97%;
    }
}
@media screen and (max-width: 566px) {
    input[type="button"] {
        margin-right: 0;
    }
}

```

Логіка роботи генератора (код javascript)

```

$(document).ready(function(){

    $('#submit').click(function(){

        var user_quantity = $('#quantity').val();
        var bit_in = $('#bit').val();
        var module = $('#module').val();
    });
}

```

```

var coefficient = $('#coefficient').val();

if(user_quantity.length<=0 || bit_in.length<=0 || module.length<=0 || coefficient.length<=0){
    alert("Заповніть всі поля");
    return;
}

$("html").niceScroll();
$('.output-con').show();
$('.output-con .container').html("");

var quantity = [];
var quantity_mod = [];
var quantity_res = [];
var sindrom = [];

var bit_s = Math.log2(module)+1;
bit_s = Math.round(bit_s);
var bit = bit_in-1;
coefficient = coefficient.split(',');

for (var i = 1; i<=user_quantity; i++) {
    quantity.push("b0" + i);
    quantity_mod.push("b" + i);
    quantity_res.push("\tno_error_" + i + "(" + quantity_mod[i-1] + "," + quantity[i-1] + ",s);\n")
}

$('.output-con .container').append('<div class="output">'+
    '<pre>module korec_3m ('+quantity+', '+quantity_mod+',s);\n'+
    '\tinput ['+bit+':0] '+quantity+;\n'+
    '\tinput wire ['+bit_s+':0] s;\n'+
    '\toutput wire ['+bit+':0]'+quantity_mod+;\n'+
    '<br>'+
    quantity_res.join(" ") +
    '<br>endmodule<br></pre>'+

```

```
'</div>');  
  
var bit_length;  
  
switch (bit_in){  
  case '3':  
    bit_length = 7;  
    break;  
  case '4':  
    bit_length = 15;  
    break;  
  case '5':  
    bit_length = 31;  
    break;  
  case '6':  
    bit_length = 63;  
    break;  
  case '7':  
    bit_length = 127;  
    break;  
  case '8':  
    bit_length = 255;  
    break;  
}  
  
var bit_length_to = bit_length - (bit_length * 2);  
for (var j = 0; j <= quantity_res.length-1; j++) {  
  var kof = coefficient[j];  
  for (var k = bit_length; k >= bit_length_to; k-- ){  
    var kn = k;  
    var znak;  
    if(kn<0){  
      znak = '+';  
      kn = -kn
```


ДОДАТОК Ж

Акти впровадження результатів роботи

**ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ
«ТЕРНОПІЛЬСЬКЕ КОНСТРУКТОРСЬКЕ БІУРО РАДІОЗВ'ЯЗКУ «СТРІЛА»**

46020, м. Тернопіль, вул 15 Квітня, 6, тел/факс – 28-75-00, 28-72-00, tkbr_strila@ukr.net, tkbr_strila@mail.ru
р/р 26009308101054 в ТБВВ 10019/08 Філія Тернопільського обласного управління АТ "Ощадбанк",
МФО 338545, код 14042350
№ 46 від 12.02.2016 р.

ЗАТВЕРДЖУЮ

Директор ТОВ «СТРІЛА»
О.О. Рафалюк
02 2016 р.
АКТ

**впровадження результатів дисертаційної роботи Яцківа Василя
Васильовича “Теоретичні основи створення і структурна організація
компонентів безпроводних сенсорних мереж підвищеної ефективності”,
представленої на здобуття наукового ступеня доктора технічних наук**

Даний акт складений у тому, що результати дисертаційної роботи Яцківа В.В. “Теоретичні основи створення і структурна організація компонентів безпроводних сенсорних мереж підвищеної ефективності” використані при вдосконаленні комплексу дистанційного керування технологічними процесами „СТРІЛА-М”, зокрема:

- метод виправлення багатократних помилок на основі модулярних коректуючих кодів;
- схемотехнічні реалізації синтезованих на програмованих логічних інтегральних схемах пристроїв кодування/декодування даних на основі модулярних коректуючих кодів для виправлення двохкратних символічних помилок;
- структура протоколу передавання даних з використанням коректуючих кодів.

Застосування вказаних методів, схемотехнічних рішень та протоколу передавання даних дозволило підвищити надійність передавання даних та зменшити час реакції системи при виникненні аварійних ситуацій в системах контролю електричних підстанцій.

Головний інженер



Піскун С. О.



АКТ

впровадження результатів дисертаційної роботи Яцківа Василя Васильовича “Теоретичні основи створення і структурна організація компонентів безпроводних сенсорних мереж підвищеної ефективності”, представленої на здобуття наукового ступеня доктора технічних наук

Даний акт складений у тому, що результати дисертаційної роботи Яцківа В.В. “Теоретичні основи створення і структурна організація компонентів безпроводних сенсорних мереж підвищеної ефективності” використані при розробці наступного покоління безпроводних модулів CyFi (CYRF7936) фірми «Cypress Semiconductor», зокрема:

– модулярні коректуючі коди, які характеризуються високою швидкістю процесів кодування/декодування даних, забезпечують виявлення і виправлення помилок при виконанні арифметичних операцій та можливість побудови адаптивних схем контролю помилок в безпроводних сенсорних мережах;

– генератор Verilog коду для синтезу на програмованій логічній інтегральній схемі декодера модулярних коректуючих кодів по заданих параметрах, що дозволило підвищити швидкість розробки та виключити помилки при написанні коду.

Фірмі «Cypress Semiconductor» передано алгоритми завадостійкого кодування/декодування даних на основі модулярних коректуючих кодів та програмне забезпечення генератора Verilog коду.

Впровадження перерахованих вище результатів дисертаційного дослідження Яцківа В.В. при розробці безпроводних модулів, дозволило підвищити надійність передавання даних та зменшити затрати енергії на передачу даних за рахунок реалізації адаптивної схеми контролю помилок

Консультант фірми
Cypress Semiconductor
к. т.н., доцент
vick@cypress.com
тел. +380667726780

Креміль В.Т.

ЗАТВЕРДЖУЮ
 Директор ТОВ «Яваре»
 Череватий О.П.
 « 10 » лютого 2016 р.

АКТ

впровадження результатів дисертаційної роботи
 Яцківа Василя Васильовича “Теоретичні основи створення і структурна
 організація компонентів безпроводних сенсорних мереж підвищеної
 ефективності”, представленої на здобуття наукового ступеня
 доктора технічних наук

Результати дисертаційної роботи Яцківа В.В., які направлені на підвищення ефективності роботи розподілених комп’ютерних систем, зокрема, безпроводних сенсорних мереж, використані в розробках нашої компанії.

Запропоновані автором дисертації підходи до підвищення надійності та живучості розподілених систем, при розробці концепції побудови безпроводних сенсорних мереж, на основі принципів колективного інтелекту дозволили підвищити надійність функціонування розподіленого програмного забезпечення систем моніторингу.

Розроблені методи та протоколи передачі даних на основі мережного кодування використовуються при розробці інтелектуальної системи клімат контролю великих офісних приміщень.

Впровадження даних результатів дозволило підвищити надійність роботи систем моніторингу за рахунок використання децентралізованого управління на основі принципів колективного інтелекту мурашиних колоній.

Виконавчий директор



Савчишин Р.В.


«ЗАТВЕРДЖУЮ»
 Директор КП «Тернопіль Інтеравія»
 І.Л.Сороколіт.
 09 вересня 2015 р.

АКТ

впровадження результатів дисертаційної роботи
 Яцківа Василя Васильовича «Теоретичні основи створення і
 структурна організація компонентів безпроводних сенсорних мереж
 підвищеної ефективності», представленої на здобуття наукового
 ступеня доктора технічних наук

Даним актом підтверджується, що результати дисертаційної роботи Яцківа В.В. на тему «Теоретичні основи створення і структурна організація компонентів безпроводних сенсорних мереж підвищеної ефективності» використовуються для підвищення ефективності роботи системи централізованого відеоспостереження міста Тернополя» (СЦВМТ).

В процесі розробки і впровадження системи відеоспостереження на базі технології безпроводних мультимедійних сенсорних мереж були одержані особисто Яцків В.В. і використані на підприємстві КП «Тернопіль Інтеравія» такі наукові результати:

1) метод кодування зображень в системі залишкових класів, який забезпечує підвищення швидкодії процесу кодування/декодування за рахунок поділу зображення на частини та паралельної передачі окремих частин зображення на основі багатопляхової маршрутизації;

2) метод мережного кодування даних, який базується на перетворенні системи залишкових класів та забезпечує підвищення загальної пропускної здатності безпроводної мультимедійної сенсорної мережі за рахунок розділення пакетів даних на підпакети з використанням взаємно простих модулів різної розрядності та об'єднання пакетів даних у спільних вузлах для їх подальшого передавання.

Впровадження перерахованих вище результатів дисертаційної роботи Яцківа В.В. в системі відеоспостереження на базі безпроводних сенсорних мереж дозволило підвищити надійність та загальну пропускну здатність безпроводної мережі в середньому на 60%.

Головний інженер



С.І. Журовський

«ЗАТВЕРДЖУЮ»

Перший проректор
Тернопільського національного
економічного університету

 Шинкарик М.І.
_____ 2016 р.

АКТ

впровадження у навчальний процес результатів дисертаційної роботи
Яцківа Василя Васильовича
“Теоретичні основи створення і структурна організація компонентів
безпроводних сенсорних мереж підвищеної ефективності”
на здобуття наукового ступеня доктора технічних наук

Даний акт складений про те, що результати отримані в дисертаційній роботі Яцківа В.В., зокрема: метод мережного кодування даних на основі перетворення системи залишкових класів; метод виправлення багатократних помилок на основі модулярних коректуючих кодів; метод контролю помилок при виконанні основних арифметичних операцій; метод кодування та передавання зображень в системі залишкових класів; метод передавання даних на основі стрибкоподібної зміни частоти; метод підвищення надійності передавання мультимедійних даних в безпроводних сенсорних мережах; структурні та схемотехнічні рішення пристроїв завадостійкого кодування/декодування даних використовуються в навчальному процесі факультету комп'ютерних інформаційних технологій Тернопільського національного економічного університету при підготовці магістрів та спеціалістів спеціальності «Спеціалізовані комп'ютерні системи» при викладанні дисциплін «Проектування реконфігурованих апаратно-орієнтованих процесорів на програмованих логічних інтегральних схемах», «Схемотехніка спецпроцесорів в комп'ютерних системах» та бакалаврів напрямку підготовки «Комп'ютерна інженерія» і «Комп'ютерні науки» при викладанні дисциплін «Системи передавання даних», «Програмування логічних матриць», «Безпроводні комунікаційні системи».

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



М.П. Дивак

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



Я.М. Николайчук

Завідувач кафедри інформаційно -
обчислювальних систем і управління,
д.т.н., професор



А.О. Саченко

“ЗАТВЕРДЖУЮ”

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



12 2015 р.

АКТ

про використання результатів докторської дисертації
Яцківа Василя Васильовича на тему **“Теоретичні основи створення і структурна
організація компонентів безпроводних сенсорних мереж підвищеної ефективності”**

Комісія у складі: голови – декана факультету комп’ютерних інформаційних технологій, д.т.н., проф. Дивака М.П. та членів: директора науково-дослідного інституту інноваційного розвитку та державотворення, д.е.н., проф. Длугопольського О.В., начальника відділу прогнозування і маркетингу Кушніра О.Р. склали цей акт про те, що дослідження та результати дисертаційної роботи Яцківа В.В. використані під час виконання науково-дослідних робіт на кафедрах спеціалізованих комп’ютерних систем та інформаційно-обчислювальних систем і управління 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), (відповідальний виконавець), у якій автором розроблено методи підвищення ефективності роботи безпроводних сенсорних мереж;

2) науково-дослідної роботи на тему “Безпроводні мультимедійні сенсорні мережі на основі модулярної арифметики та кодів Галуа для систем відео спостереження”, (номер державної реєстрації 0113U004321), (відповідальний виконавець), у якій автором розроблено методи підвищення надійності передавання даних в безпроводних сенсорних мережах;

3) науково-дослідної роботи на тему “Розробка теоретичних засад методів формування та цифрового опрацювання даних у розподілених спеціалізованих комп’ютерних системах”, (номер державної реєстрації 0112U008458), у якій автором розроблено концепцію побудови безпроводних сенсорних мереж на основі колективного інтелекту;

4) науково-дослідної роботи на тему “Розробка теорії та комп’ютерних засобів спеціалізованих комп’ютерних систем на основі теоретико-числових базисів Крестенсона-Галуа”, (номер державної реєстрації 0106U012530), у якій автором розроблено модифікований метод передавання даних на основі стрибкоподібної зміни частоти;

5) науково-дослідної роботи на тему “Розробка теорії, методології побудови та технічних засобів спеціалізованих комп’ютерних систем”, (номер державної реєстрації 0102U005764), у якій автором розроблено теоретичні основи та схемотехнічні рішення спецпроцесорів обробки даних в системі залишкових класів.

Голова комісії

декан факультету комп’ютерних
інформаційних технологій,
д.т.н., проф.

Дивак М.П.

Члени комісії:

директор НДІ ІРД

Длугопольський О.В.

начальник відділу ПМ

Кушнір О.Р.