

КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

УДК 681.326

**СИСТЕМА ЗАВАДОСТІЙКОГО
КОДУВАННЯ ДАНИХ НА ПЛІС**
*ФІЛІППЕНКО І.В., КУЛАК Е.М.,
СЕРГІЄНКО В.І.*

Розробляється завадостійка система бездротової передачі даних із застосуванням блокового кодування та реалізацією на ПЛІС. Пропонується спосіб реалізації алгоритму LDPC-кодування на ПЛІС, а також алгоритм декодування та схема LDPC-декодера з реалізацією на ПЛІС.

Ключові слова: VHDL-модель, синтез, коди з низькою щільністю перевірок на парність, згорткове кодування, програмно-апаратна реалізація.

Key words: VHDL-model, synthesis, low density pairing check codes, convolutional coding, implementation.

1. Вступ

В епоху сучасних технологій постійно збільшується об'єм інформації, який завжди передається через канали зв'язку. В цих каналах постійно виникають перешкоди, викликані різними обставинами, наприклад погодні явища, випромінювання від різних технічних засобів, відображення. Ці перешкоди псують якість зв'язку і призводять до втрати корисної інформації.

У сучасних системах передачі інформації широко використовуються три методи кодування: первинне, економне і завадостійке [1, 2]. *Первинне кодування* інформації дозволяє перетворювати дискретні величини у кодові комбінації. Прикладами таких кодів є міжнародний телеграфний код МТК-2, коди обміну інформацією ЯКІ-7, ЯКІ-8. Первинні коди зазвичай представляють у вигляді таблиць. *Економне кодування* дозволяє скоротити надмірність коду, іншими словами, стиснути дані. Такий метод кодування враховує статистичні характеристики джерела повідомлень і дає можливість представити знаки повідомлення в середньому меншим числом кодів символів. Наприклад, значно скоротити надмірність переданих повідомлень можна, якщо символам повідомлень, що найчастіше зустрічаються, поставити у відповідність коротші кодові комбінації, а менш імовірним символам – довші. Операцію скорочення надмірності джерела повідомлень називають економним кодуванням. Економне кодування широко використовується в цифрових системах передачі інформації для стиснення мовних, факсимільних і телевізійних повідомлень, які мають велику надмірність. Первинне і економне кодування називають також кодуванням джерела.

Завадостійке кодування дозволяє виявляти і виправляти помилки, що виникають при передачі повідомлень по каналу зв'язку. Завадостійке кодування здійснюється шляхом введення до складу переданих кодів символів надлишкової інформації (наприклад, перевірочних символів). Операцію введення надмірності для підвищення завадостійкості каналу зв'язку називають канальним кодуванням. Завадостійке кодування широко застосовується в системах зв'язку, в ЕОМ (електронна обчислювальна машина), в цифровій аудіо- і відеотехніці.

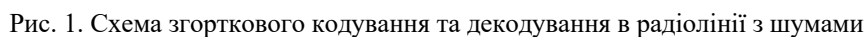
Теорія завадостійкого кодування базується на результатах досліджень, проведених Клодом Шенноном [1]. Він сформулював теорему для дискретного каналу з завадами: при будь-якій швидкості передачі двійкових символів, меншою, ніж пропускна здатність каналу, існує такий код, при якому ймовірність помилкового декодування буде як завгодно мала. Побудова такого коду досягається ціною введення надмірності, тобто застосовуючи для передачі інформації код, який має надмірності в кодових комбінаціях, можна підвищити стійкість каналу зв'язку перед перешкодами. Такі коди називають надмірними або коригуючими. Коригуючі властивості надлишкових кодів залежать від правил побудови цих кодів і параметрів коду (тривалості символів, числа розрядів, надмірності).

В даний час найбільша увага приділяється двійковим рівномірним коригуючим кодам [3]. Вони мають хороші коригуючі властивості і їх реалізація досить проста.

Найбільш часто застосовуються блокові коди. При використанні блокових кодів цифрова інформація передається у вигляді окремих кодів блоків рівної довжини. Кодування і декодування кожного блоку здійснюється незалежно один від одного, тобто кожній букві повідомлення відповідає блок з t символів. Блоковий код називається рівномірним, якщо n (значність) залишається однаковою для всіх букв повідомлення. Розрізняють роздільні і нероздільні блокові коди. При кодуванні роздільними кодами кодові операції складаються з двох роздільних частин: інформаційної та перевірочної. Інформаційні та перевірочні розряди у всіх кодових комбінаціях роздільного коду займають одні й ті ж позиції.

При кодуванні нероздільними кодами розділити символи вихідної послідовності на інформаційні та перевірочні неможливо.

Безперервними називаються такі коди, в яких введення надлишкових символів у послідовності інформаційних символів здійснюється безперервно, без поділу її на незалежні блоки.



В останні роки зусиллями вітчизняних та зарубіжних вчених освоєні методи коригуючого кодування, які забезпечують енергетичний виграш 6-7дБ в каналах з постійними параметрами та Гаусовим шумом, що дозволило підвищити завадостійкість систем зв'язку з космічними апаратами. Із теорії інформації відомо, що ефективними є достатньо довгі коди з параметрами, подібними до параметрів випадкового шуму [3]. При цьому з ростом довжини коду складність алгоритму декодування (апаратна або програмна складність декодера) значно зростає. Тому прак-

Цей алгоритм кодування дозволить реалізувати завадостійку систему передачі даних. Він також може бути реалізований на ПЛІС (програмовані логічні інтегральні схеми), що дозволить знизити затримку при кодуванні та декодуванні інформації.

- 1) аналіз та порівняння методів кодування даних;
- 2) вибір методу завадостійкого кодування;
- 3) адаптація обраного алгоритму кодування для реалізації його на ПЛІС;

- 4) розробка алгоритму кодування/декодування для реалізації його на ПЛІС;
- 5) розробка VHDL-моделі кодеку (VHDL – (Very high speed integrated circuits) Hardware Description Language – мова опису апаратури високошвидкісних інтегральних схем), верифікація вихідної моделі кодеку та його моделі після імплементації;
- 6) реалізація пристрою на мікросхемі ПЛІС;
- 7) проведення діагностичного експерименту.

2. Порівняльний аналіз ефективності згорткового кодування та LDPC-кодування

Для моделювання згорткового кодування в каналі зв'язку з QPSK модуляцією (Quadrature Phase Shift Keying – квадратурна фазова маніпуляція) була побудована функціональна модель в програмному пакеті MATLAB Simulink (рис. 1). Моделювання завадостійкого LDPC-кодування даних в радіолінії з QPSK модуляцією виконувалося також за допомогою програмного пакета MATLAB. Щоб закодувати інформаційну послідовність, потрібно перемножити її з породжуючою матрицею G обраного варіанту пари LDPC матриць (перевірочна та породжуюча). Перевірочні та породжуючі матриці для довжин інформаційного слова 128, 1024 та 4096 біт були рекомендовані CCSDS [4, 5].

В умовах, коли передавач знаходиться далеко від приймача та має обмежену потужність, з якою може передавати повідомлення, дуже важливу роль грає кодування та енергетичний виграв, який воно може забезпечити при застосуванні. Графік залежності імовірності бітової помилки з різними видами кодування показаний на рис. 2.

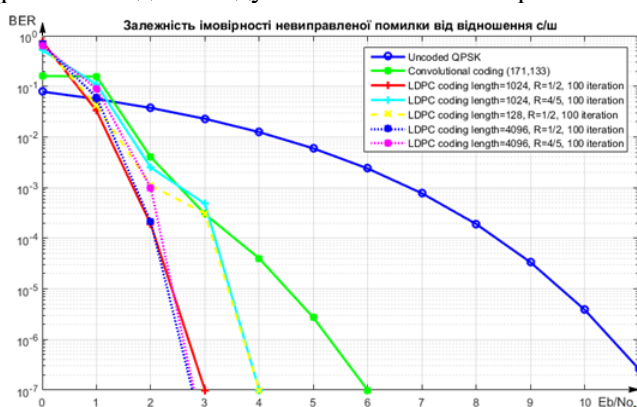


Рис. 2. Графік для порівняльного аналізу різних видів кодування

З рис. 2 видно, що при застосуванні в каналі зв'язку LDPC-кодування показники бітової помилки при тих самих значеннях співвідношення сигнал/шум кращі, навіть при жорсткому декодуванні, ніж при згортковому кодуванні. Моделювання LDPC-кодування проводилося для довжин інформаційного слова 128, 1024 та 4096 біта, а з ростом довжини інформаційного слова залежність бітової помилки від відношення сиг-

нал/шум при застосуванні LDPC-кодів прямує до межі Шеннона.

Отже, виходячи з вище зазначеного, використання LDPC-кодування дозволяє реалізувати систему передачі даних з достатньо малою ймовірністю помилки на біт, і відповідно може бути застосоване в радіолініях зв'язку з космічними апаратами. Через це для програмно-апаратної реалізації було вибрано саме LDPC-кодування з довжиною коду 128.

3. Розробка структури і vhdл-моделі LDPC-кодеку

Розроблено модуль кодера, оснований на алгоритмі кодування запропонованому в [4]. Цей алгоритм наведений на рис. 3. Він якнайкраще підходить для використання на ПЛІС, оскільки використовує зсувні регістри, що дозволяє значно збільшити швидкість кодування порівняно з кодуванням, коли використовуються ресурси процесора.

Породжуючу матрицю для цього коду згенеровано таким чином, що для відтворення матриці на кодері не обов'язково зберігати всю матрицю. Достатньо зберігати кожен шістнадцятий рядок матриці, які називаються циркулянтами. Інші рядки, а саме 60 ($15 \cdot 4$), можна відтворити через використання циклічного зсуву над рядком циркулянта, як наведено на рис. 3. Для цього використовується такий модуль.

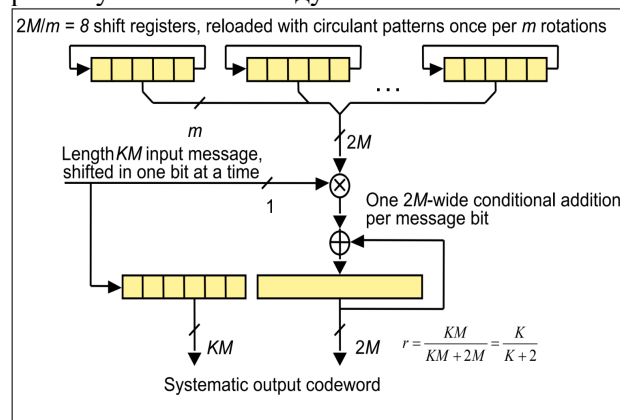


Рис. 3. Алгоритм LDPC-кодування даних

Існує чотири простих зсувних регістри. Ці регістри виконують операцію циклічного зсуву вправо. Вони об'єднані в один регістр, що називається shift_register. Цей регістр має три вхідні сигнали: clk, load та load_vector. Перший з них робить зсув даних, як показано на рис. 3. Load та load_vector дозволяють загрузити нові дані (циркулянти) в цей зсувний регістр. Для цього необхідно передати дані по шині load_vector, а потім подати строб даних. Таким чином, використовуючи цей зсувний регістр, можна не зберігати всю породжуючу матрицю, а лише перший, сімнадцятий, тридцять третій та сорок дев'ятий рядки цієї матриці. Для зберігання цих рядків використовується пакет f_file_encoder. Для

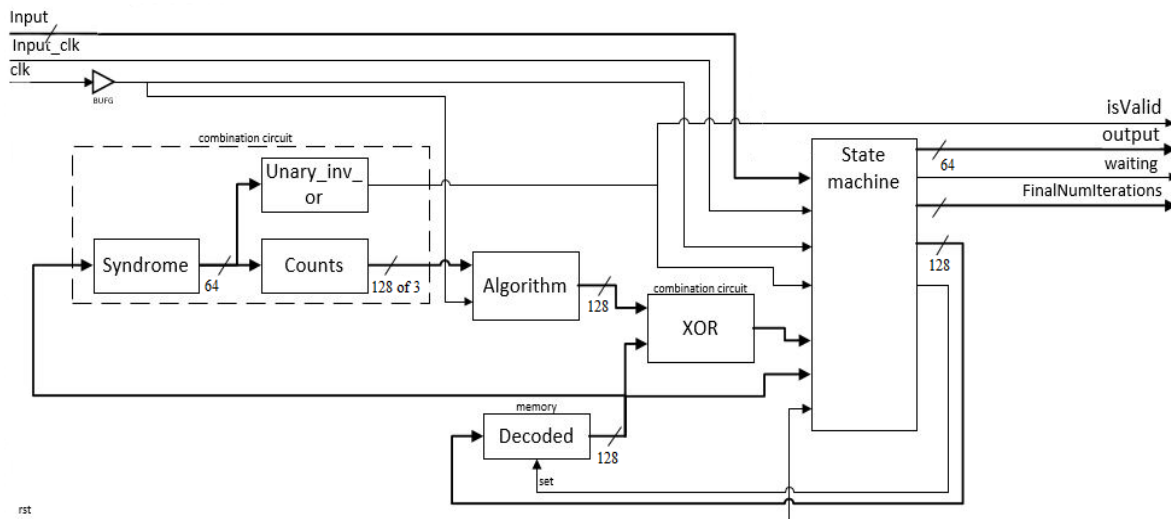


Рис. 4. Схема декодера

відтворення усіх інших рядків використовується зсувний регістр `shift_register`.

Маючи спосіб відтворити усю породжуючу матрицю, є можливість виконувати кодування. Для цього інформація, що потрапляє на кодер, відразу потрапляє як інформаційна частина (див. рис. 3) кодового слова (інформації, яка буде передаватися по каналу зв'язку). Далі перший біт інформаційного слова помножується на перший циркулянт. Результат записується в кодову частину кодового слова, а циркулянт записується в зсувний регістр, для подальшого формування породжуючої матриці. Потім проводиться операція зсуву зсувного регістра і проводиться множення вже другого інформаційного біта на значення зсувного регістра. Результат складається по модулю два з кодовою частиною кодового слова, що вже збережений в кодері.

Такі операції проводяться ще 14 разів. Потім відбувається зміна циркулянта і операції повторюються. Після проходження всіх циркулянтів буде отримана закодована інформація (кодове слово), яка складається з інформаційної та кодової частин.

Було також розроблено схему декодера, яка наведена на рис. 4. Функціонує вона так. На приймальній стороні встановлюють LDPC-декодер, в якому записана відповідна перевірна матриця для декодування прийнятого кодового слова. Вхідна послідовність з демодулятора, після якого значення кодової послідовності є двійковими, разом зі стробом, який сигналізує про надходження послідовності на вхід декодера, поступає на механізм станів. Цей механізм станів записує вхідну послідовність до внутрішньої пам'яті `decoded` та починає очікувати закінчення першої ітерації декодування. У цей час дані з внутрішньої пам'яті поступають на вхід блоку `syndrome`. У блоці відбувається перевірка ко-

вого слова на парність з використанням перевіркової матриці.

Цей модуль дозволяє виявити, чи є у вхідного кодового слова хоча б одна не пройдена перевірка на парність. Якщо не пройдено перевірки, то вихідний сигнал модуля `unary_inv_or` буде дорівнювати нулю, якщо ж усі перевірки на парність пройдені, то вихідний сигнал буде мати значення одиниці.

У цей же час значення проходження перевірок на парність з модуля `syndrome` потрапляють на лічильники не пройдених перевірок. Для кожного біта кодового слова створюється свій лічильник який у двійковому вигляді відображає кількість не пройдених перевірок на парність для даного біта.

В результаті цієї операції можна буде знайти біт, який не проходить максимальну кількість перевірок.

Після цього значення виходів модулів `counts` потрапляють в модуль `algorithm`. Цей модуль відповідає за вибір біта для інверсії в процесі декодування. В запропонованій реалізації був використаний алгоритм `Bif Flipping`. Його головною ідеєю є вибір біта, який не пройшов максимальну кількість перевірок на парність, із всіх бітів кодового слова, і повинен бути інвертований.

Для вибору цього біта використовується послідовне порівняння кількості не пройдених перевірок на парність для кожного біта. Для оптимізації цього процесу і підвищення швидкості однієї ітерації декодування можна використати паралельний пошук максимального числа не пройдених перевірок, реалізувавши це комбінаційною схемою, але це понесе за собою велику кількість витрачених ресурсів ПЛІС.

В результаті блок `algorithm` генерує вектор виправлення помилок, який на позиціях, що треба інвертувати, виставляє одиницю, а на позиціях, які потрібно залишити, без змін ставить нуль. Цей вектор потрапляє на наступний блок (див. рис. 4), в якому побітно виконується операція хог. Результат цієї операції передається в механізм станів і у кінці кожної ітерації перезаписується у внутрішню пам'ять. Ітерації повторюються до моменту повного виправлення всіх помилок, або до того моменту, коли кількість ітерацій перевищить поріг, заданий `generis`-константою.

Для верифікації моделі кодера було використано середовище Active-HDL 9.1, в якому проведено симуляцію роботи кодера (рис. 5). На вхід тактування було подано меандр з частотою 10 МГц, на вхід даних випадкове слово з 64-х біт, а на вхід `input_clk` було подано строб. Результат кодування було перевірено за допомогою MATLAB.

Для перевірки роботи декодера також було використане середовище Active-HDL 9.1, в якому проведено симуляцію роботи декодера (рис. 6). На вхід тактування було подано меандр з частотою 10 МГц, на вхід даних спочатку було подано нульове кодове слово та строб. Як видно з рис. 6, декодер відразу визначив, що подане на нього кодове слово є правильним, та вийшов з циклу декодування з нульовою кількістю ітерацій.

Після цього на вхід даних декодера було подане слово з однією помилкою та строб даних. Як видно з рис. 6, декодер визначив, що кодове слово не є вірним (сигнал `IsValid` дорівнює нулю), та почав цикл декодування з виправленням помилок. Після першої ітерації декодер сформував вектор виправлення помилки, в якому правильно виставивши біт виправлення, зміг відновити первісну інформацію і вивести слово без помилок.

Оскільки пристрій, який проектується, в основному складається з елементів пам'яті (тригерів), синтез його було виконано на FPGA (Field-Programmable Gate Array – програмована користувачем вентиля матриця). Хоча CPLD (Complex Programmable Logic Device – складний програмований логічний пристрій) мають більш високу швидкодію, їх використовують переважно для реалізації логічних функцій великого числа змінних. Оскільки для синтезу та імплементації використовувалося інструментальне середовище Xilinx Project Navigator V5.1, то вибір мікросхеми FPGA проводився з елементної бази фірми Xilinx.

В результаті було виконано синтез і імплементацію, створені файли – опису пристрою після імплементації. Загальна кількість використаних ресурсів при синтезі кодера наведена на рис. 7.

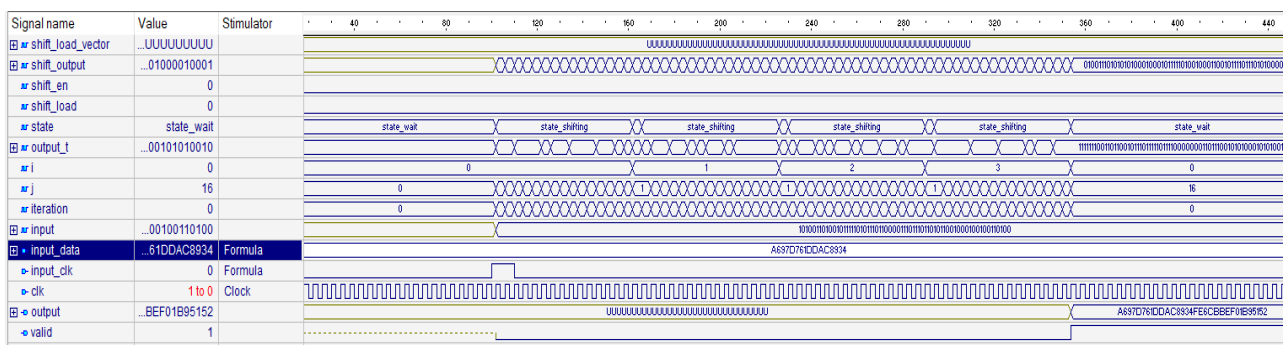


Рис. 5. Часова діаграма роботи кодера

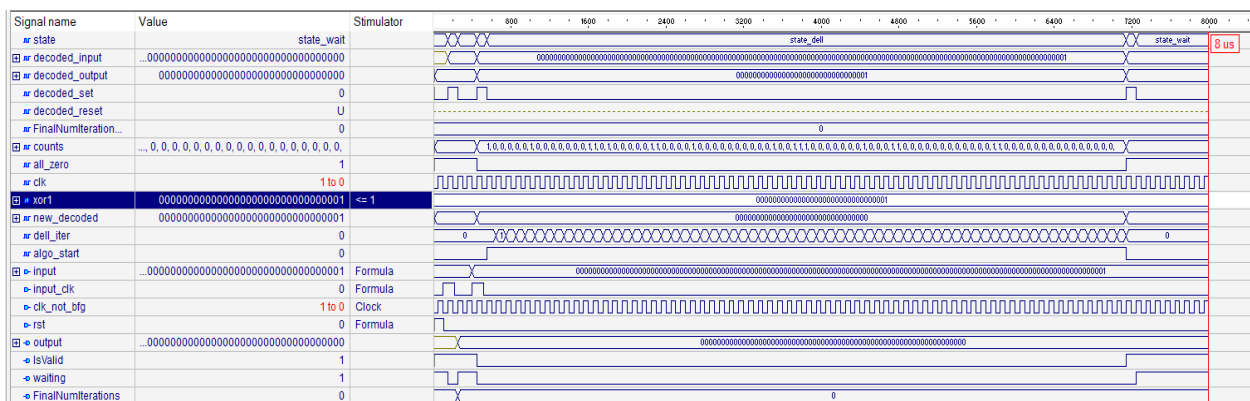


Рис. 6. Часова діаграма роботи декодера

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	259	0	53200	0.49
LUT as Logic	259	0	53200	0.49
LUT as Memory	0	0	17400	0.00
Slice Registers	410	0	106400	0.39
Register as Flip Flop	345	0	106400	0.32
Register as Latch	65	0	106400	0.06
F7 Muxes	8	0	26600	0.03
F8 Muxes	4	0	13300	0.03

Рис. 7. Використані ресурси при синтезі кодера
Загальна кількість використаних ресурсів для синтеза декодера наведено на рис. 8.

Було також виконано моделювання кодеку після імплементації, та его прошивка на кристалі XILINX XC7Z035-2LFBG676i.

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	750	0	53200	1.41
LUT as Logic	750	0	53200	1.41
LUT as Memory	0	0	17400	0.00
Slice Registers	528	0	106400	0.50
Register as Flip Flop	528	0	106400	0.50
Register as Latch	0	0	106400	0.00
F7 Muxes	163	0	26600	0.61
F8 Muxes	78	0	13300	0.59

Рис. 8. Використані ресурси при синтезі декодера

4. Діагностичний експеримент

Перед проведенням діагностичного експерименту були поставлені дві мети. Першим етапом необхідно було перевірити працездатність розроблених модулів кодера і декодера в реальній радіолінії. Метою другого етапу було отримати якісний аналіз роботи даних модулів.

Для проведення першого етапу діагностичного експерименту використовувалися дві плати PicoZed SDR AD9361 Development Kit з вбудованими приймачами Analog Devices AD9361 з плоскими антенами і системами на кристалах XILINX XC7Z035-2LFBG676i, які містять двоядерні ARM Cortex-A9 процесори і ПЛІС (FPGA). В даному експерименті використовувалися приймачі та ПЛІС, на яких були прошиті розроблені модулі LDPC-кодера і декодера. Експеримент проводився в лабораторних умовах без використання підсилювачів потужності і з використанням аттенуаторів для додаткового зниження рівня потужності сигналу, що передається.

У процесі підготовки до проведення діагностичного експерименту була побудована схема, яка наведена на рис. 9.

У схемі діагностичного експерименту є два пристрої діагностування. Пристрій 1 складається з генератора інформації 1, LDPC-кодера і модулятора.

Генератор інформації 1 відповідає за формування потоку заздалегідь відомих пакетів даних, які надходять на LDPC-кодер. У модулі LDPC-кодера відбувається безпосередньо кодування інформації, яка надалі передається на модулятор, після чого поступає в радіоканал.

Пристрій 2 складається з демодулятора, LDPC-декодера, генератора інформації 2 і модуля порівняння. Демодулятор демодулює дані з радіоканалу і передає їх на LDPC-декодер і модуль порівняння. Розроблений модуль LDPC-декодера виконує декодування і передає пакети даних на модуль порівняння. В цей же час генератор інформації 2 генерує пакети інформації, в тій же послідовності, що і модуль генерації інформації 1, та передає їх на модуль порівняння. Все це дозволяє зробити порівняння трьох потоків даних: ідеального (з генератора інформації 2), з помилками після радіоканалу (з демодулятора) і з виправленими помилками (з LDPC-декодера). Порівняння потоків у даному експерименті дозволяє визначити, чи виникають помилки в радіоканалі і чи справляється LDPC-декодер з ними.

Для проведення аналізу отриманих результатів було проведено ряд експериментів з різними відносинами сигнал/шум. Для імітації різних радіоканалів використовувалася зміна відстані між антенами пристроїв 1 і 2. Було проведено два експерименти, в яких антени знаходилися на відстані 10 см і 10 м відповідно. Перший експеримент показав, що дані, які відправлялися з пристрою 1, потрапляли через радіоканал в пристрій 2 без спотворень. Інформація на модулі порівняння збігалася для всіх трьох потоків даних.

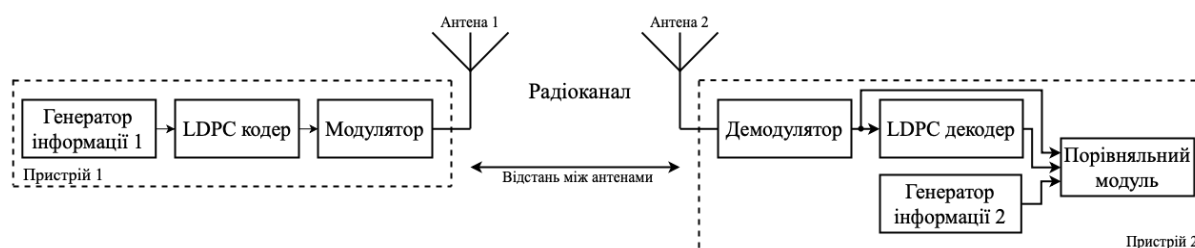


Рис. 9. Схема діагностичного експерименту (етап 1)

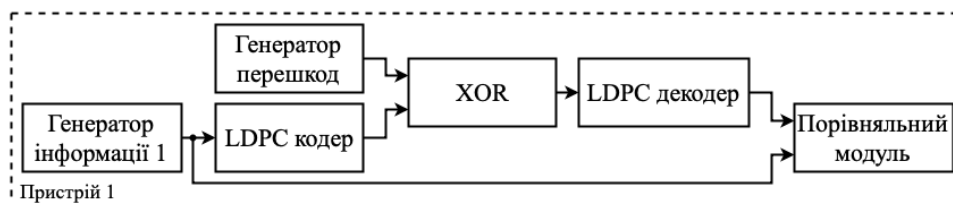


Рис. 10. Схема діагностичного експерименту (етап 2)

Таблиця 1

Кількість помилок, доданих в переданий пакет інформації	Загальна кількість проведених тестів (пакетів) для даної кількості доданих помилок	Кількість виправлених пакетів	Кількість не виправлених пакетів	Відсоток виправлених пакетів
10	52	38	14	73%
11	40	26	14	65%
12	39	21	18	54%
13	36	17	19	47%
14	33	7	26	21%
15	32	5	27	16%
16	30	3	27	10%

Таблиця 2

Кількість помилок, доданих в переданий пакет інформації	Загальна кількість проведених тестів (пакетів), для даної кількості доданих помилок	Кількість виправлених пакетів	Кількість не виправлених пакетів	Відсоток виправлених пакетів
8	18	10	8	56%
13	19	8	11	42%
16	18	8	10	44%
19	18	7	11	38%
23	18	6	12	33%
28	18	2	16	11%
29	18	2	16	11%

Другий експеримент (рис. 10) показав, що інформація, яка була передана через радіоканал, спотворювалася, але LDPC-декодер виконував виправлення помилок. В середньому в радіоканалі виникала одна помилка на 1000 біт.

Для якісного аналізу роботи розроблених пристроїв був проведений другий етап діагностичного експерименту. Метою було визначити, скільки помилок в прийнятих пакетах здатний виправити LDPC-декодер. Метою також було визначити, чи є різниця в результатах роботи залежно від відносного розташування помилок в пакеті. Для проведення даного експерименту була побудована така схема (див. рис. 10).

Відмінністю даної схеми від попередньої є те, що експеримент проводився всередині одного пристрою. Для імітації передачі даних по радіоканалу використовувався такий підхід. Дані з генератора інформації подавалися на LDPC-кодер, на якому відбувалося кодування пакетів. Після цього до сформованого пакету за допомогою операції суми по модулю два додавалися помилки, згенеровані модулем генерації помилок. Тільки після цього дані з помилками подавалися на LDPC-декодер, в якому відбувалося декодування і передача отриманих даних на модуль порівняння. В результаті використання даного підходу під

час проведення експерименту контролювалася кількість і відносне розташування помилок в переданих пакетах. Експеримент проводився для різної кількості помилок в пакеті.

При аналізі результатів експериментів було виділено дві основні групи помилок в прийнятих пакетах: помилки, розташовані в пакетах випадковим чином, та помилки, розташовані поспіль (пакетні помилки). Аналіз показав, що дані групи помилок необхідно розглядати окремо, оскільки вони показують різні статистичні дані. Результати експериментів щодо роботи декодера при випадковому розташуванні помилок в прийнятому пакеті наведені в табл. 1. Результати експериментів щодо роботи декодера при пакетному розташуванні помилок в прийнятому пакеті наведені відповідно в табл. 2.

Якщо порівняти значення відсотка виправлених пакетів для двох виділених груп появи помилок, то можна помітити, з якогось моменту (а саме від 13 помилок в пакеті) LDPC-декодер краще справляється з пакетними помилками, ніж з випадково освіченими (рис. 11). Це можна пояснити тим, що для декодування використовуються перевірочні матриці з малою кількістю перевірок на парність і ці перевірки розташовані псевдовипадковим чином.

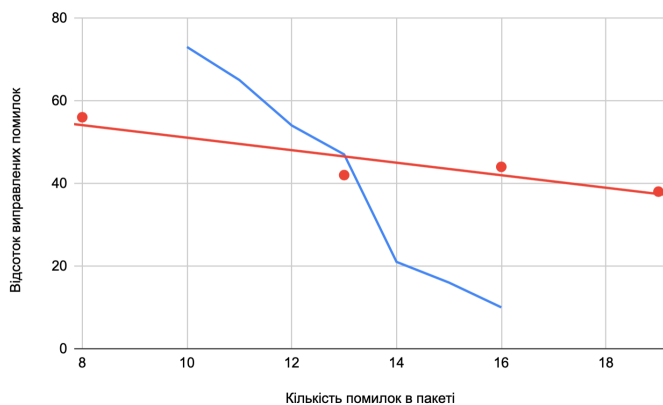


Рис. 11. Порівняння результатів для пакетних і випадкових помилок

Це дозволяє отримувати інформацію для декодування певного біта пакета не тільки від сусідніх біт, але і від далеко розташованих бітів. Дана особливість дозволяє не використовувати перемежувач і деперемежувач в схемі радіолінії, що є економією апаратних і часових ресурсів всього пристрою.

Проведений діагностичний експеримент показав таке: 1) спроектовані модулі LDPC-кодера і декодера виконують свою функціональну задачу, а саме забезпечують стійкість під час передачі інформації; 2) LDPC-коди краще справляються з пакетними помилками, ніж з випадково розташованими.

5. Висновки

В результаті роботи було виконано:

- 1) вибір методу завадостійкого кодування;
- 2) реалізація на ПЛІС (FPGA) обраного алгоритму кодування;
- 3) розробка алгоритму декодування для реалізації його на ПЛІС;
- 4) розробка VHDL-моделі кодеку, верифікація вихідної моделі кодеку, моделі кодеку після імплементації;
- 5) проведення діагностичного експерименту.

Наукова новизна: 1) Отримано подальший розвиток алгоритму LDPC-кодування для реалізації його на ПЛІС шляхом використання зсувних регістрів і суматорів та керуючого автомата. 2) Розроблено алгоритм та схеми LDPC-декодера на ПЛІС з використанням зсувних регістрів і суматорів, виконуючого та керуючого автомата, який синхронізує роботу блоків пристрою.

Практична значущість: розроблені модулі можуть бути використані для реалізації завадостійкого LDPC-кодування інформації у високошвидкісних системах передачі даних, а саме відеоданих із супутників, дронів, безпілотників. Результати роботи можна використовувати як готові модулі при розробці різноманітних пристроїв, в яких використовується завадостійке кодування, наприклад, в радіолінії зв'язку з космічними апаратами.

Література: 1. Прокис Д. Цифровая связь / Д. Прокис. М.: Радио и связь, 2000. 800 с. 2. Волков Л.Н. Системы цифровой радиосвязи: базовые методы и характеристики / Л.Н. Волков, М.С. Немировский, Ю.С. Шинаков: М.: Эко-Трендз, 2005. 392с. 3. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение / Р. Морелос-Сарагоса. М.: Техносфера, 2005. 320 с. 4. Experimental specification. Short block length LDPC codes for TC synchronization and channel coding / Consultative Committee for Space Data Systems 01.04.2015. W.: Space data systems, 2015. 39 p. 5. TM synchronization and channel coding: Recommended standard 131.0-B-2. 01.08.2011. W.: Standard USA, 2011. 93 p. 6. Radio frequency and modulation systems – part 1 earth stations and spacecraft: Recommended standard 401.0-B. 01.12.2013. W.: Standard USA, 2013. 244 p.

Transliterated bibliography:

1. Prokis D. Cifrovaja svjaz' / D. Prokis. M.: Radio i svjaz', 2000. 800 s.
2. Volkov L.N. Sistemy cifrovoj radiosvjazi: bazovye metody i harakteristiki / L.N. Volkov, M.S. Nemirovskij, Ju.S. Shinakov: M.: Jeko-Trendz, 2005. 392s.
3. Morelos-Saragosa R. Iskustvo pomehoustojchivogo kodirovanija. Metody, algoritmy, primenenie / R. Morelos-Saragosa: M.: Tehnosfera, 2005. 320s.
4. Experimental specification. Short block length LDPC codes for TC synchronization and channel coding / Consultative Committee for Space Data Systems 01.04.2015. W.: Space data systems, 2015. 39 p.
5. TM synchronization and channel coding: Recommended standard 131.0-B-2. 01.08.2011. W.: Standard USA, 2011. 93 p.
6. Radio frequency and modulation systems – part 1 earth stations and spacecraft: Recommended standard 401.0-B. 01.12.2013. W.: Standard USA, 2013. 244 p.

Надійшла до редколегії 11.12.2019

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Філіппенко Інна Вікторівна, канд. техн. наук, доцент кафедри АПОТ ХНУРЕ. Наукові інтереси: проектування цифрових пристроїв на базі мікроконтролерів, цифрові фільтри. Адреса: Україна, 61166, Харків, пр. Науки, 14, тел. 702-13-26.

Кулак Ельвіра Миколаївна, канд. техн. наук, доцент кафедри АПОТ ХНУРЕ. Наукові інтереси: автоматизоване проектування цифрових автоматів, тестопридатне проектування. Адреса: Україна, 61166, Харків, пр. Науки, 14, тел. 702-13-26.

Сергієнко Владислав Ігорович, студент ХНУРЕ. Наукові інтереси: автоматизоване проектування цифрових пристроїв. Адрес: Україна, 61166, Харків, пр. Науки, 14, тел. 702-13-26.

Filippenko Inna Victorovna, PhD, Associate Professor, Design Automation Department, KNURE. Scientific interests: design based on microcontrollers, digital filters. Address: Ukraine, 61166, Kharkiv, Nauka Avenue, 14, tel. 702-13-26.

Kulak Elvira Nikolaevna, PhD, Associate Professor, Design Automation Department, KNURE. Scientific interests: automated design of digital machines, HDL. Address: Ukraine, 61166, Kharkiv, Nauka Avenue, 14, tel. 702-13-26.

Serhiienko Vladyslav Igorovich, student, KNURE. Scientific interests: automated design of digital units. Address: Ukraine, 61166, Kharkiv, Nauka Avenue, 14, tel. 702-13-26.