КОМП'ЮТЕРНІ СИСТЕМИ ТА КОМПОНЕНТИ

УДК 004.056 : 004.424.47

Ю. В. Баришев, В. М. Запасна

МЕТОД ТА ІНТЕРФЕЙСИ ДЛЯ ПЕРЕДАВАННЯ ДАНИХ В ЛІНІЯХ З ВИСОКИМ РІВНЕМ ЗАВАД

Вінницький національний технічний університет, м. Вінниця

Анотація. Розглянуто сучасні інтерфейси передавання даних. Наведено новий метод передавання даних в лініях з високим рівнем завад, який дозволяє забезпечити цілісність цих даних. Даний метод реалізовано у вигляді синхронного та асинхронного інтерфейсів. Для самосинхронізації передавання в асинхронному режимі передбачено використання групового кодування, яке реалізується відповідним структурним блоком мікропроцесорної системи. Контроль автентичності даних, отриманих протягом сеансу зв'язку, здійснюється за рахунок гешування, яке реалізується в інтерфейсах окремим блоком. Наведено результати компонентного тестування інтерфейсів, отриманих внаслідок комп'ютерного моделювання з використанням мови опису апаратури VHDL.

Ключові слова: мікропроцесор, завади, передавання даних, синхронний інтерфейс, асинхронний інтерфейс.

Аннотация. Рассмотрены современные интерфейсы передачи данных. Приведен новый метод передачи данных в линиях с высоким уровнем помех, который позволяет обеспечить целостность этих данных. Данный метод реализован в виде синхронного и асинхронного интерфейсов. Для самосинхронизации передачи в асинхронном режиме предусмотрено использование группового кодирования, которое реализуется соответствующим структурным блоком микропроцессорной системы. Контроль аутентичности данных, полученных во время сеанса связи, совершается за счет хеширования, реализованного в интерфейсах отдельным блоком. Приведены результаты компонентного тестирования интерфейсов, полученные вследствие компьютерного моделирования с использованием языка описания аппаратуры VHDL.

Ключевые слова: микропроцессор, помехи, передача данных, синхронный интерфейс, асинхронный интерфейс.

Abstract. The modern interfaces for data transfer are considered. New method for data transmission via lines with high interference level is proposed which allows to ensure the integrity of the data. This method is implemented as synchronous and asynchronous interfaces. Group encoding is implemented as appropriate structural microprocessor unit for self-synchronization providing for the asynchronous transfer mode. The control of data received at the transfer session authenticity is performed by hashing, which is implemented by the special interface block. Results of interfaces components testing are presented, which are gotten by computer modeling using hardware description language VHDL.

Keywords: microprocessor, interferences, data transfering, synchronous interface, asynchronous interface.

Вступ

Широке поширення електронних пристроїв разом з пришвидшенням роботи обробки даних привнесло низку нових задач. Однією з таких задач є забезпечення цілісності даних при передаванні даних. Це пов'язано з тим, що на багатьох підприємствах та організаціях лінії передавання даних проходять через приміщення з великою кількістю електричних кабелів, які створюють електромагнітні завади. Ці завади впливають на цілісність даних, що передаються такими лініями. Тому постає необхідність для розробки мікропроцесорної системи передавання даних, що допоможе зберегти цілісність даних, які передаються лініями з високим рівнем завад. Особливо проблема завад постає у силовій електроніці, адже вплив завад може порушити роботу всієї системи, а водночас силові пристрої, будучи джерелом завад, самі ж від них і страждають.

Актуальність

Передавання даних критичне для переважної більшості сучасних інформаційних систем, які передбачають розподілення обчислень між різними вузлами цих систем. При цьому такі обчислення можуть бути як безпосередньо задані користувачем, так і "допоміжними" (системними) – такими, що забезпечують виконання процесів, які вже розв'язують задачі задані користувачем. Така організація обробки даних дозволяє збільшити продуктивність системи за рахунок розпаралелення обчислень та/або використання обчислювальних вузлів, що мають спеціалізовані блоки, розроблені для розв'язання певних задач. Водночас це потребує забезпечення постійного обміну даними між різними вузлами.

Особливої значущості це набуває для інформаційних систем критичних інфраструктур, де висуваються підвищенні вимоги щодо забезпечення цілісності та автентичності інформації. Методи передавання даних, які використовуються в сучасних інтерфейсах мікроконтролерів, в більшості випадків можуть забезпечити відповідність цим вимогам, однак їх використання стає проблематичних за умов впливу завад на лінії зв'язку, що зокрема актуально для електроенергетичних систем, які відносяться до критичних інфраструктур. Використання додаткового кодування, що реалізується програмними засобами, які розв'язують ці задачі на мережевому, транспортному, сеансовому або, навіть, прикладному рівнях породжує необхідність у передаванні додаткових технічних даних для кожного з протоколів, що зменшує інформаційну швидкість обміну даними. З цього випливає актуальність розв'язку задачі забезпечення цілісності та автентичності даних при їх передаванні лініями зв'язку, в яких внаслідок завад підвищена

ймовірність виникнення помилок в кадрі порівняно зі стандартними вимогами QoS для інтерфейсів, саме на рівні апаратних засобів передавання.

Мета

Метою даного дослідження є покращення захищеності при передаванні інформації шляхом розробки мікропроцесорної системи передавання даних.

Задачі

Для досягнення мети необхідно розв'язати такі задачі:

1. Проаналізувати сучасні методи передавання даних, що використовуються в мікропроцесорних системах.
2. Розробити метод передавання даних, що підвищуватиме захист цілісності та автентичності даних, що передаються.
3. Розробити інтерфейси, що реалізують запропонований метод.

Аналіз відомих методів передавання даних, що використовуються в мікропроцесорних інтерфейсах

Одним з найпоширеніших методів асинхронного передавання в мікропроцесорних системах є метод, використаний в універсальному асинхронному приймачі-передавачі – інтерфейсі UART. Цей метод забезпечує напівдуплексний зв'язок і передбачає передавання даних по 8 біт в кадрах розміром в 10 або 11 бітів. Для забезпечення синхронізації даний метод передбачає обрамлення інформаційних даних стартовим та стоповим бітами, які сигналізують про ініціацію та коректне завершення передавання кадру відповідно [1]. Крім того більшість реалізацій UART передбачають автоматичний контроль цілісності даних методом контролю бітової парності. Коли ця функція ввімкнена, після останнього біту дописується біт, що містить інформацію про парність кількості одиничних біт в даному кадрі [1]. Цей метод забезпечує виявлення однократної помилки і може, за певних умов, дозволити виявити помилки більшої кратності, але даний метод не надає можливості виправити виявлену помилку. Це робить необхідним організацію запиту на повторне передавання даних, що відповідно до розміру кадру має бути 10 або 11 біт, яке за умови високого рівня завад може також спотворитись і потребуватиме, в свою чергу, формування на іншій сторони запиту на повторне передавання запиту на повторне передавання і так далі. Відповідно за умов високого рівня завад такий метод зіб'ється на постійне передавання запитів на повторне передавання даних замість надсилання власне даних, які потребують цього передавання (зручно уявити цю ситуацію при ймовірності некоректного прийому біта даних 0,1), що неприйнятно для передавання даних в межах критичних інфраструктур. Вищевикладена критика справедлива й для методів передавання, які будуються на даному, зокрема, метод, використаний в інтерфейсі Octal UART [2].

В методах передавання, які використовуються в інтерфейсах SPI та SSI забезпечується дуплексний зв'язок та синхронізація пристроїв, однак не передбачається підтвердження прийому даних з боку пристроїв, які працюють в режимі slave, тому дані можуть передаватись безадресно [3, 4]. Також дані методи не передбачають процедури виявлення помилок, а тому порівняно з інтерфейсом UART забезпечують менший рівень стійкості до завад.

Метод передавання даних, що використовується інтерфейсом I^2C, передбачає підтвердження приймання даних. Останнє, в свою чергу, надає можливість організації повторного передавання даних, які не були прийняті адресатом. Крім того, цей метод передбачає одночасну взаємодію декількох пристроїв в режимі master та протокол арбітражу, який попереджає виникнення колізій в шині при одночасному початку передавання даних декількома пристроями [5]. Проте такий метод передавання не дозволяє ні виявляти та виправляти помилки в кадрах, ні автентифікувати дані. Відповідно при виникненні помилки в кадрі, вона не буде виявленою, що не дозволяє адресату, приймаючи дані, визначити їх коректність.

Інтерфейс USB передбачає використання циклічних кодів для виявлення помилок і, у випадку виникнення такої події, повторного пересилання даних [6]. Причому залежно від виду трафіка може використовуватися CRC кодування різного типу. Крім того для поліпшення характеристик самосинхронізації цей інтерфейс використовує групове кодування 128b/132b. однак відсутність можливості виправлення помилки в кадрі при високому рівні завад може спричинити те, що для коректного передавання даних може знадобитися неоднократне повторення процедури передавання даних. Відповідно це негативно вплине на загальну інформаційну швидкість цього інтерфейсу.

Ще одним методом, який забезпечує передавання даних в мікропроцесорній техніці, є метод, що використовується в інтерфейсі CAN. Для досягнення безпеки CAN використовує поточний контроль (передатчики порівнюють рівні бітів, що були передані, з рівнями на шині), побітове заповнення та перевірку кадру повідомлення. Вузли CAN можуть відрізняти тимчасові помилки від постійних відмов [7]. Однак при виявленні помилок не передбачено жодного механізму її виправлення.

Таким чином, методи передавання даних, які використовуються в сучасних інтерфейсах не передбачають методів, які забезпечують виявлення та виправлення помилок, а як наслідок їх апаратну підтримку. Саме це робить необхідним "огортання" даних методів передавання іншими методами на вищих

рівнях моделі відкритих систем OSI, що породжує або збільшення технічної інформації, а також відсутність використання протоколами вищих рівнів всіх можливостей інтерфейсів щодо завадостійкості, або – необхідність у повторному передаванні даних, що, як показав аналіз, не може вважатись адекватним розв'язанням задачі передавання даних в лініях з високим рівнем завад.

Метод передавання даних

Для інтеграції інтерфейсів, що пропонуються в даному дослідженні, до існуючих систем передавання пропонується залишати існуючі способи адресації пристроїв. Для цього інтерфейси будуть імітувати для пристроїв роботу їх адресатів. Наприклад, якщо розглядати систему обміну інформацією між двома пристроями (Пристрій 1 та Пристрій 2), що здійснюють обмін за допомогою інтерфейсу SPI [3] (рис. 1).

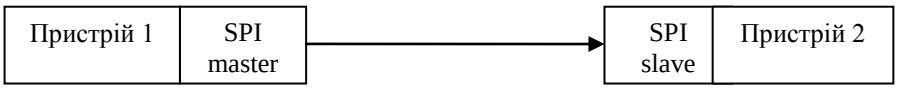

Рисунок 1 – Приклад системи обміну інформацією

Інтеграцію до даної системи пропонується здійснювати так, як це наведено на рис. 2.

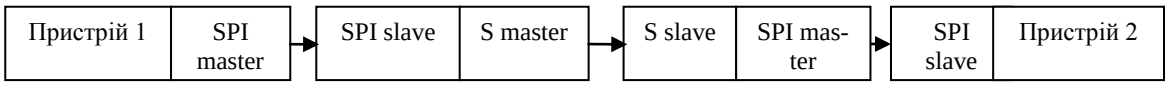

Рисунок 2 – Приклад використання інтерфейсу в лініях з високим рівнем завад

Як видно з рис. 1, при вбудовуванні пристроїв, що пропонуються, для програмного забезпечення Пристрою 1 та Пристрою 2 не доведуться вносити ніяких змін, оскільки як до інтеграції Пристрій 1 здійснював обмін з пристроєм, що має інтерфейс SPI slave, так і після неї він здійснює обмін з таким пристроєм. Аналогічним чином використання додаткових інтерфейсів не спричинятиме необхідність зміни програмного забезпечення Пристрою 2. На рис. 3 та 4 наведено алгоритми передавання та приймання даних відповідно, що формалізують запропонований метод передавання, у випадку асинхронного режиму передавання.

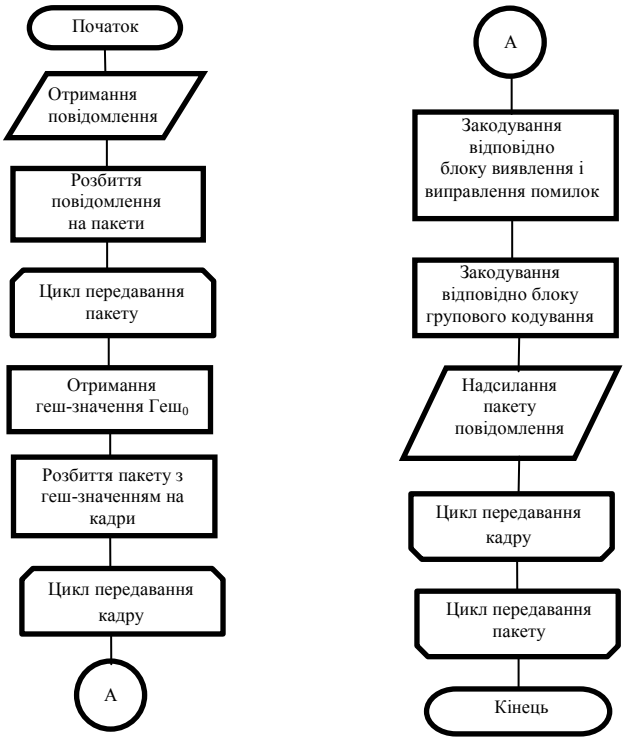

Рисунок 3 – Алгоритм передавання даних в асинхронному режимі в лініях з високим рівнем завад

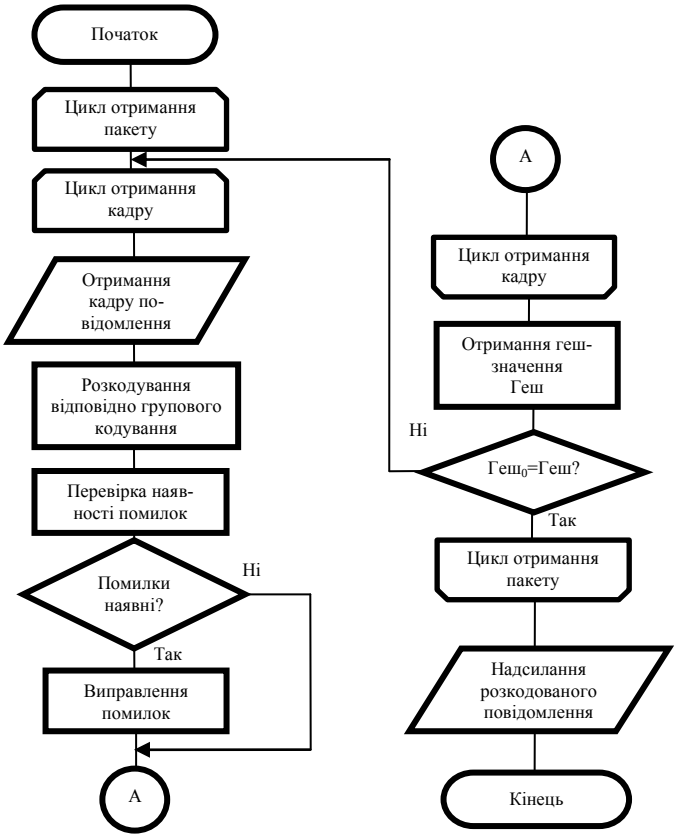

Рисунок 4 – Алгоритм отримування даних в асинхронному режимі в лініях з високим рівнем завад

У випадку синхронного режиму передавання алгоритми, що формалізують метод, мають аналогічний вигляд, однак у цьому випадку не виконується процедура закодування/розкодування відповідно до групового коду, який покликаний покращити показники самосинхронізації, а тому в синхронному режимі передавання його використання недоцільне.

Як видно з рис. 3 та 4 даний метод передбачає розбиття повідомлення на пакети для кожного з яких обчислюється геш-значення для забезпечення захисту автентичності та цілісності повідомлень. Апаратна реалізація гешування підвищеної швидкості для аналогічних задач розглядалася зокрема в роботах [8, 9]. Пакет із дописаним до нього геш-значенням відповідно до запропонованого методу, розбивається на кадри. В межах кадрів використовується кодування, що забезпечує виявлення та виправлення помилок, а також, у випадку асинхронного передавання, групове кодування [10, 11].

В табл. 1 наведено теоретичні оцінки ймовірності виявлення та виправлення помилок в кадрах при використанні запропонованого методу за умови різного рівня завад.

Таблиця 1 – Теоретичні оцінки ймовірності виявлення та виправлення помилок в кадрах при використанні запропонованого методу

Ймовірність помилки отримання 1 біта	Ймовірність наявності помилки в кадрі	Ймовірність однократної помилки в кадрі	Ймовірність багатократної помилки в кадрі	Частка коректно прийнятих кадрів
Синхронний режим передавання				
0,01	0,113615	0,1074	0,006215	0,945298
0,03	0,306158	0,2575	0,048658	0,841069
0,06	0,52408	0,3645	0,15958	0,695505
0,09	0,677525	0,3827	0,294825	0,56485
0,15	0,857758	0,3012	0,556558	0,351148
0,21	0,90758	0,2434	0,66418	0,268186

Продовження табл. 1

Ймовірність помилки отримання 1 біта	Ймовірність наявності помилки в кадрі	Ймовірність однократної помилки в кадрі	Ймовірність багатократної помилки в кадрі	Частка коректно прийнятих кадрів
Асинхронний режим передавання				
0,01	0,139942	0,129	0,010942	0,92181
0,03	0,366749	0,2575	0,109249	0,702115
0,06	0,604708	0,3645	0,240208	0,60277
0,09	0,756992	0,3827	0,374292	0,505554
0,15	0,912646	0,3012	0,611446	0,330029
0,21	0,970866	0,1885	0,782366	0,194157

Як видно з табл. 1, при ймовірності помилки в біті 0,01, ймовірність отримання кадру, що мітить помилку становитиме близько 0,1 і з цих 10% кадрів понад 92% будуть виправлені. При ймовірності помилки 0,1 запропонований метод, на відміну від розглянутих вище, дозволятиме коректно приймати половину зі спотворених завадами кадрів. При отриманні оцінок, наведених в табл. 1, враховувалась можливість коду виправляти лише однократні помилки в межах кадру. Очевидно, що при використанні параметрі кодів, які дозволяють виправляти помилки більшої кратності, оцінки даного методу покращаться.

Інтерфейси передавання даних лініями з високим рівнем завад

Для реалізації запропонованого методу було реалізовано інтерфейси для синхронного та асинхронного режимів передавання даних. На рис. 5 наведено структуру мікропроцесорного пристрою для передавання даних в лініях з високим рівнем завад.

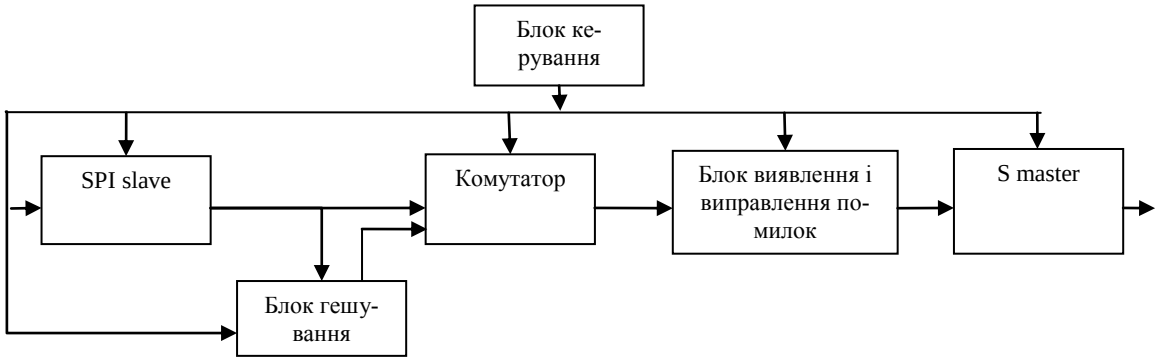

Рисунок 5 – Пристрій передавання з синхронним інтерфейсом S Master

Дані, що надсилаються з SPI slave, передаються на комутатор та паралельно проходять через блок гешування. Геш-значення передається останнім пакетом [9]. Після комутатора данні потрапляють на блок виявлення та виправлення помилок, де доповнюються даними, які дозволять у випадку виникнення помилок їх виправити після приймання на іншій стороні зашумленої лінії. Після чого дані потрапляють на синхронний інтерфейс S master, який призначений для синхронного передавання розширених в даному пристрої пакетів даних .

Аналогічним чином запропоновані ідеї можуть бути використані для асинхронного методу передавання. Однак в цьому випадку після блоку виявлення та виправлення помилок з метою забезпечення самосинхронізації даних пропонується використовувати блок групового кодування.

Для практичної реалізації запропонованого методу було розроблено синхронний та асинхронний інтерфейси, які використовували код Хемінга для виявлення та виправлення помилок, код ГК 4/5. На рис.6 наведено структуру асинхронного інтерфейсу.

З рис. видно, що інтерфейс має 17-розрядний регістр зсуву. Така кількість розрядів обумовлена тим, що відповідно до методу кодування за Хемінгом [10, 11] для 8 інформаційних біт необхідно обчислити 4 перевірочних біти. Отримана внаслідок такого кодування група з 12 бітів розбивається на 3 частини по 4 біти кожна. І, внаслідок кодування груповим кодом ГК 4/5, кожна група розширюється до кодового слова довжиною 5 біт. Таким чином, 12 бітів, які надходять на блок групового кодування перетворюються у 15 біт. Для організації асинхронного способу передавання попереду цих 15 біт додається біт "старт", а наприкінці біт "стоп". Таким чином, до іншого пристрою буде надсилатись 17 біт, що обумовлює розрядність регістру зсуву в інтерфейсі AS.

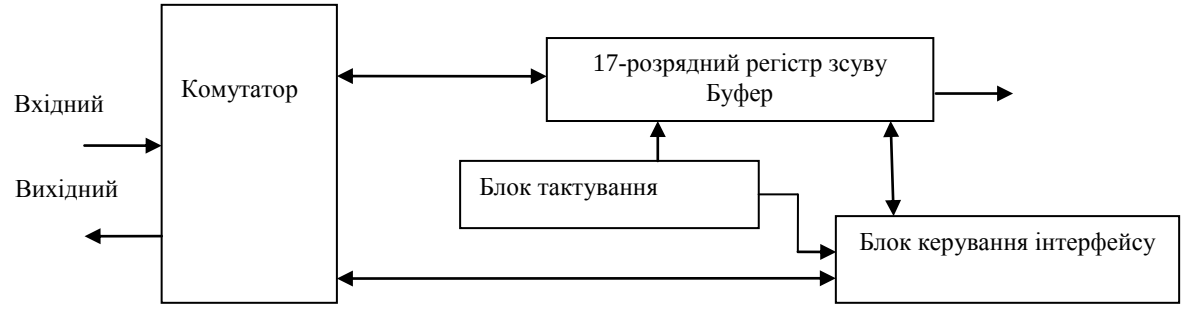

Рисунок 6 – Схема асинхронного інтерфейсу AS

Роботу даного інтерфейсу було досліджено за допомогою комп’ютерного моделювання у середовищі ModelSim Starter Edition. Для його реалізації були описані архітектури:

- GK_encoding_arc, яка передбачає наявність одного процесу, який виконуватиметься при зміні сигналу start (для реалізації сутності GK_encoding).
- GK_decoding_arc, яка передбачає наявність одного процесу, який виконуватиметься при зміні сигналу start (для реалізації сутності GK_decoding).
- Hamming_encoding_arc, яка передбачає наявність одного процесу, який виконуватиметься при зміні сигналу start (для реалізації сутності Hamming_encoding).
- Hamming_decoding_arc, яка передбачає наявність одного процесу, який виконуватиметься при зміні сигналу start (для реалізації сутності Hamming_decoding).

Результати тестування роботи блоку розкодування ГК 4/5 наведено в табл. 2. Внаслідок тестування роботи блоку закодування ГК 4/5 отримано вихідні дані, які відповідають результатам перекодування в код ГК-4/5 наведених в таблиці 2.

Таблиця 2 – Результати тестування розкодування ГК 4/5

№	Вхідні дані	Вихідні дані
1	inputData="10101"	outputData="0010" error='0'
2	inputData="01001"	outputData= "1011" error='0'
3	inputData="10111"	outputData= "0100" error='0'
4	inputData="00001"	outputData="1111" error='1'

Під час тестування роботи блоку розкодування ГК 4/5, коли надходять дозволені комбінації, то відбувається їх розкодування та надсилання на блок керування сигнала error='0', що сигналізує про успішне завершення роботи. Коли ж надходять заборонені комбінації, то надсилається сигнал error='1' та вихідні дані рівні "1111".

Під час роботи блоку закодування за Хемінгом вхідні дані inputData закодовуються відповідно до формул коду Хемінга, що підтверджують вихідні дані outputData, отримані при тестуванні даного блоку. Результати тестування роботи блоку розкодування за Хемінгом наведено в таблиці 3.

Таблиця 3 – Результати тестування блоку розкодування за Хемінгом

№	Вхідні дані	Вихідні дані
1	inputData="000000000000"	outputData="00000000" error='0' warning='0'
2	inputData="000000001000"	outputData= "10000000" error='0' warning='1'
3	inputData="001110010011"	outputData= "00111001" error='0' warning='1'
4	inputData="010101001111"	outputData= "00000000" error='1' warning='0'

Тестування роботи розкодування за Хемінгом дало підтвердження виправленню однієї помилки та виявленню багатьох помилок.

Під час першого випадку тестування подано дані без помилок, що підтвердили відправлені на блок керування сигнали error, warning. В наступних двох випадках присутня одна помилка, тому відбулось виправлення та відправлено про це сигнал warning='1' на блок керування. В останньому тестуванні відбулась трикратна помилка, тому inputData не були розкодованими, про що повідомляє сигнал error='1'.

Висновки

Аналіз відомих методів передавання даних в мікропроцесорних системах дозволив виявити необхідність у розробці методів та апаратних засобів передавання, які б забезпечували підвищення завадостійкості за умов наявності високого рівня завад у лінії зв'язку.

Для розв'язання поставленої задачі запропоновано використовувати метод передавання даних та нові інтерфейси на його основі. Цей метод за рахунок введення надлишковості в дані дозволив підвищити рівень захищеності цілісності та забезпечити захист автентичності цих даних. Високий рівень надлишковості даних (46,7 % у кодовому слові) робить його недоцільним при використанні в системах з високими вимогами до швидкості передавання, однак в системах з високим рівнем завад і меншими вимогами щодо швидкості передавання, зокрема системи керування електроенергетичними системами, використання запропонованого методу є доцільним, оскільки він забезпечує підвищену швидкість передавання порівняно з відомими методами за рахунок зменшення необхідності у повторному пересиланні кадрів. Відповідно до отриманих теоретичних оцінок останнє виконується за умов ймовірності спотворення одного біта даних понад 3%, що обумовлюється більшою ймовірністю у необхідності повторного передавання кадру та ймовірністю виникнення помилок у такому запиті щодо повторного надсилання кадру. Запропонований метод реалізовано у вигляді синхронного та асинхронного інтерфейсів, які реалізовано мовою VHDL та проведено комп'ютерне моделювання роботи їх та їхніх складових, що дозволило підтвердити отримані теоретичні оцінки.

Список літератури

1. KeyStone Architecture : Universal Asynchronous Receiver/Transmitter (UART). User Guide / Texas Instruments. Literature Number: SPRUGP1, November 2010 [Електронний ресурс]. – Режим доступу: http://www.ti.com/lit/ug/sprugp1/sprugp1.pdf – Назва з екрану.

2. Enhanced octal universal asynchronous receiver/transmitter (Octal UART). Data Sheet / Philips Semiconductors. SCC2698B, 07 Aug. 2006 (Supersedes data of 2000 Jan 31) [Електронний ресурс]. – Режим доступу: http://cache.nxp.com/documents/data_sheet/SCC2698B.pdf?pspll=1 – Назва з екрану.

3. Новицкий А. Синхронный последовательный интерфейс SPI в микроконтроллерах «от А до Я» и его реализация на примере ADuC70xx фирмы Analog Devices / А. Новицкий // Компоненты и технологии [Електронний ресурс]. – Режим доступу: http://www.kit-e.ru/articles/interface/2009_03_53.php – Назва з екрану.

4. Novak P. SSI - Interface and protocol for industrial sensors / Petr Novak // XXVI. ASR '2001 Seminar, Instruments and Control, Ostrava, April 26 - 27, 2001 [Електронний ресурс]. – Режим доступу: http://akce.fs.vsb.cz/2001/asr2001/Proceedings/papers/50.pdf – Назва з екрану.

5. UM10204. I^2C-bus specification and user manual / NXP Semiconductors. Rev. 6 — 4 April 2014. [Електронний ресурс]. – Режим доступу: http://www.nxp.com/documents/user_manual/UM10204.pdf – Назва з екрану.

6. USB 3.1 Specification. Revision. 1.0, July 26, 2013 / [Hewlett-Packard Company and others] [Електронний ресурс]. – Режим доступу: http://www.usb.org/developers/docs// – Назва з екрану.

7. CAN Specificatioin ver. 2.0 / Robert Bosch GmbH, 1991. – 72 p. [Електронний ресурс]. – Режим доступу: http://www.bosch-semiconductors.de/media/ubk_semiconductors/pdf_1/canliteratur/can2spec.pdf – Назва з екрану.

8. Баришев Ю. В. Методи та засоби швидкого багатоканального гешування даних в комп'ютерних системах : монографія / Ю. В. Баришев, В. А. Лужецький. — Вінниця : ВНТУ, 2016. — 142 с.

9. Лужецький В. А. Апаратні засоби для реалізації багатоканального керованого хешування. / В. А. Лужецький, Ю. В. Баришев. // Системи обробки інформації.– 2011. – №3. – С. 130–133.

10. Азаров О.Д. Аналого-цифрові інтерфейси ОЕМ: навч. Посіб. Для студ. вузів/ О.Д. Азаров, В.П. Марценюк., Н.О. Біліченко. – Вінниця: УНІВЕРСУМ – Вінниця, 2006 р. – 180с.

11. Кузьмин И.В. Основы теории информации и кодирования. 2-е узд./И.В. Кузьмин, В.А. Кедрус. – К.: "Вища школа", 1986 г. – 238 с.

Стаття надійшла: 10.10.2016

Відомості про авторів

Баришев Юрій Володимирович – к. т. н., доцент кафедри захисту інформації, ВНТУ.
Запасна Валентина Миколаївна – магістрант кафедри захисту інформації, ВНТУ.