

ПРОГРАМНЕ ТА АПАРАТНЕ ТЕСТУВАННЯ ДЕКОДЕРУ TURBO-PRODUCT-КОДІВ**Я.М. Крайник, В.О. Перов**Чорноморський національний університет імені Петра Могили,
вул. 68 Десантників, 10, Миколаїв, 54003, Україна; e-mail: codebreaker7@ukr.net

У роботі представлена методика тестування декодера Turbo-Product-кодів на програмному та апаратному рівнях. Дана методика дозволяє отримати результати відносно корегуючої здатності декодера, а також пропускну здатності. Метою даного дослідження є розробка уніфікованої методики, яка дозволить проводити тестування та оцінку показників декодерів для будь-яких підвидів Turbo-Product-кодів на будь-якій апаратній платформі. Проведене дослідження дозволило виявити проблемні місця при декодуванні за допомогою заданого алгоритму та встановити, при яких рівнях завад декодер буде працювати стабільно. З практичної точки зору це дозволяє визначити, який код доцільно застосовувати при заданих умовах відповідно до параметрів пропускну здатності та виправної здатності та досягти максимально ефективного показника співвідношення двох вказаних параметрів. Методика тестування передбачає проходження кількох стадій тестування. У роботі використовувались програмні засоби для моделювання ModelSim, мова програмування Python, а також апаратні засоби, серед яких варто виділити розроблений лабораторний стенд на базі двох відлагоджувальних модулів. Розроблене апаратне забезпечення у вигляді схематехнічного опису декодерів на мові VHDL дозволяє проводити тестування широкого набору кодів, які підтримуються декодером. Розроблені апаратні засоби включають до свого складу лабораторний стенд, який складається з двох відлагоджувальних плат: плати з мікросхемою ПЛІС та інтерфейсною платою для підключення та передачі і прийому даних. На базі першої плати розгортається декодер, в той час як друга призначена для того, щоб забезпечити необхідну швидкість обміну даними. Комбінація з розробленого програмного забезпечення та лабораторного стенду використовувалась при проведенні тестування за представленою у даній роботі методикою. Відповідно до розробленої методики у подальшому можливо розробити універсальний підхід для тестування апаратних рішень для завадостійких кодів.

Ключові слова: декодер, Turbo-Product-коди, програмне тестування, апаратне тестування.

Вступ

Декодери кодів прямого виправлення помилок на даний момент є центральними елементами телекомунікаційних систем. Вони є одними з найскладніших компонентів на стороні отримуючого обладнання. Від того, яку швидкість декодування здатен забезпечувати декодер, залежить загальна швидкість передачі даних у системі. Для того, щоб мати можливість представити можливі режими роботи системи та її основні характеристики на основі математичних операцій, які використовуються у ході декодування, варіантами є проведення моделювання спеціалізованими програмними засобами або організація лабораторних тестувань з залученням апаратного забезпечення. У даній роботі використовуються обидва варіанти для того, щоб підтвердити коректність розгортання моделі на ресурси програмовних логічних інтегральних схем (ПЛІС). Представлена методика, відповідно до якої проводились дослідження, а також результати проведених досліджень.

До кодів з прямим механізмом виправлення помилок відносяться і коди, які отримали назву турбо-коди-добутки (англ. Turbo-Product-Codes – TP-коди) [3]. Вони

представляють собою коди, що містять горизонтальну та вертикальну компоненти коду (рядки та стовпці). Декодування відбувається за ітераціями, і результат попередньої ітерації подається на вхід новій ітерації, тому у назві наявна частина «Turbo».

У даній роботі представлені результати оцінки системи, яка описана у попередніх працях [1, 2]. Дана система представляє собою декодер, який комбінує м'який та жорсткий підхід для декодування TR-кодів. Він дозволяє об'єднати швидкість роботи жорсткого декодера з більшою коригуючою здатністю декодера, який працює з м'якими значеннями. Перш за все, такий декодер повинен забезпечувати кращу виправну здатність, ніж декодер жорстких рішень. У якості складових компонентів TR-кодів часто використовуються SECDED-коди (SingleErrorCorrectionDoubleError Detection), які дозволяють виправити одиночну помилку та вказати на наявність подвійної помилки, проте не виправити її. Саме ці коди розглядаються у даній роботі. У ній представлені показники роботи системи за умови використання різних кодів. Проводиться перевірка роботи в різних умовах зашумленості каналу передачі даних. З практичної точки зору це дозволить визначити, які коди доцільно використовувати за різних умов роботи системи.

Тестування апаратних рішень на базі програмовних логічних інтегральних схем (ПЛІС) є складною задачею, яка потребує різних підходів в залежності від задачі, яка вирішується розробкою опису для даного типу мікросхем. Традиційним підходом є тестування на основі тестового оточення з подальшим переносом реалізації на апаратну частину. При цьому послідовно виконується декілька стадій перевірки з використанням тестового оточення: поведінкова перевірка, що призначена для перевірки коректності реалізації базових положень, перевірка на рівні Place&Route або PostFit, яка дозволяє якомога більше наблизити моделювання з використанням програмних засобів до роботи реального пристрою. Дані стадії моделювання є важливими, проте вони не дозволяють повністю гарантувати працездатність схеми при завантаженні файлу конфігурації у мікросхему. Лише перевірка роботи опису безпосередньо у мікросхемі ПЛІС дозволяє гарантувати коректність роботи пристрою.

Проте, оскільки умови тестування можуть доволі часто підлягати змінам, то розроблення готового пристрою для цього не є доцільним. Альтернативою цьому є розробка лабораторного стенду, який можна підлаштувати відповідно до змін вимог без значних затрат.

Важливим аспектом у тестуванні рішень на базі ПЛІС є розробка відповідного прикладного програмного забезпечення, яке повинне взаємодіяти з апаратним забезпеченням. Специфіка програмного забезпечення залежить від прикладної задачі, яка вирішується. Для випадку декодування завадостійких кодів, до яких відносяться TR-коди, дослідження яких проводиться у даній роботі, необхідно проводити перевірку на основі передачі зашумлених даних, прийому декодованих даних та порівняння отриманих результатів з очікуваним.

У даній роботі пропонується методика проведення дослідження показників ефективності роботи декодера TR-кодів на базі ПЛІС при тестуванні і на програмному, і на апаратному рівнях. У подальшому даний підхід може бути розширений для застосування при тестуванні всіх типів завадостійких кодів.

Основна частина

Методика проведення оцінки показників декодера передбачає наявність наступних етапів:

- генерація набору тестових даних на основі показників кодів для тестування. Відповідно, на даному етапі мають бути згенеровані дані для усіх кодів, декодування яких здатен проводити декодер;

- визначення показників зашумленості каналу передачі даних. Для представлення цього показника можуть використовуватись різні фізичні показники, які у кінцевому результаті можуть бути приведені від одного до іншого (показник сигнал/шум для біту, для символу, імовірність помилки);

- накладання шуму відповідно до обраного показника, який характеризує якість передачі даних у каналі. Крок, який здійснюється між попереднім та наступним показником для цього значення залежить від того, наскільки детальну характеристику необхідно отримати для досліджуваного декодера;

- передача зашумлених даних на вхід декодеру. У якості моделі реального декодеру, зазвичай, використовується програмне забезпечення, яке проводить однакові операції з декодером, для якого проводиться тестування;

- отримання декодованого повідомлення. На цьому етапі в якості виходу можуть використовуватись як м'які дані, так і жорсткі дані;

- аналіз результуючих даних, отриманих на виході декодеру шляхом порівняння з початковим повідомленням. На цьому етапі, в основному, використовуються жорсткі дані, а м'які дані використовуються для того, щоб простежити загальний хід процесу декодування.

На першому етапі для генерації тестових даних може використовуватись довільний двовимірний масив двійкових значень відповідної розмірності, які відповідають характеристикам коду. Після цього слова з даного масиву мають бути закодовані з використанням генеруючих матриць кодів. Розмір масиву збільшується відповідно до показників довжин коду. Для представлення передачі по каналу з шумами проводиться відображення жорстких вхідних даних на м'які. Відображення відбувається на значення, що відповідають максимальному абсолютному значенню, яке використовується у системі, з присвоєнням відповідного знаку. Таким чином, для перевірки можуть використовуватись два початкові масиви: жорстких значень і м'яких значень.

На другому етапі визначається, наскільки якісним є канал передачі і які завади присутні у каналі. Для тестування можуть використовуватись різні типи помилок, які вносяться у початковий масив даних: просте інвертування, внесення зовнішнього впливу відповідно до значення амплітуди конкретного біту та інші техніки. Таким чином, на третьому етапі у повідомленні з'являються помилки/відмінності у порівнянні з початковим кодовим словом, які декодер має виправити.

Після цього мають бути реалізовані наступні етапи, які є ключовими для проведення оцінки. Програмне забезпечення, яке повністю повторює роботу декодеру з точки зору математичних операцій, що використовуються, зазвичай, є підготовленим заздалегідь, оскільки саме на його основі проводиться попереднє тестування якості алгоритму декодування. Дане програмне забезпечення може бути як звичайною консольною утилітою, так і повноцінною програмою з графічним інтерфейсом користувача. Проте через можливі зміни, пов'язані з цільовою платформою декодеру, отримана апаратна реалізація може не завжди відповідати початковій версії програмної реалізації, тому важливим моментом є повна та цілковита відповідність операцій, які проводяться на обох платформах. Для більшої наочності та більшої інформативності дослідження доцільно проводити легування результатів окремих ітерацій декодування або, навіть, більш дрібних стадій обробки даних. Наявність такого функціоналу дозволяє проводити порівняння з результатами, які отримані у спеціалізованих засобах моделювання системи для ПЛІС на базі спеціально розроблених тестових оточень. Порівняння кожної окремої стадії обчислення є обов'язковою умовою для проведення подальших досліджень.

На виході програми отримується декодоване повідомлення у м'якому або жорсткому вигляді. Результуюча кількість помилок для випадку м'яких значень визначається на основі співпадіння знаку м'якого значення у вхідному та вихідному

повідомленнях. Відмінності у абсолютних значеннях за умови співпадіння знаку не є суттєвими.

Проведення паралельного тестування з використанням як програмного, так і апаратного забезпечення надає переваги у тому, що можна одразу проводити аналіз результатів декодування. Порівняння результатів декодування під час моделювання роботи ПЛІС є більш затратним з точки зору часу, який необхідний на моделювання. Саме тому під час виконання даного дослідження був зібраний лабораторний стенд, який складався з відлагоджувальної плати з мікросхемою ПЛІС, інтерфейсної плати для спрощення підключення між комп'ютером та платою. Структурна схема з'єднань між компонентами розробленого лабораторного стенду представлена на рисунку 1.

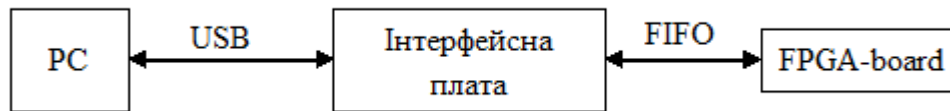


Рис. 1. Структурна схема підключення компонентів стенду

Для зв'язку з комп'ютером використовується інтерфейс передачі даних USB 2.0. Максимальна швидкість передачі даних для даної специфікації становить 480 Мбіт/с. Інтерфейсна плата, яка працює в режимі FIFO (паралельний інтерфейс передачі даних) здатна, відповідно, до документації забезпечувати швидкість передачі даних близько 40 Мбайт/с. Це означає, що дані показники є достатньо близькими, оскільки у даному випадку враховується лише корисна інформація, що передається. Мікросхема ПЛІС на відлагоджувальній платі зчитує дані з використанням блоку FIFO, а оброблені дані передаються за допомогою того самого блоку. Це пов'язано з тим, що на інтерфейсній платі наявні буфери як для прийому, так і для передачі даних.

У якості складових компонентів для побудови лабораторного стенду використовувались:

- інтерфейсна плата FT2232H MiniModule [5];
- відлагоджувальна плата Altera DE0-SoC-Nano [6].

Інтерфейсний модуль має у своєму складі в якості основного компоненту мікросхему FT2232H. Дана мікросхема здатна працювати у двох режимах FIFO (синхронний та асинхронний) для забезпечення високої пропускної здатності. У цих режимах використовується паралельний інтерфейс передачі даних. Також дана мікросхема може працювати і в якості інтерфейсу для протоколів з меншою швидкістю передачі даних – UART, SPI, I2C, JTAG. Висока пропускна здатність інтерфейсу у даному випадку означає те, що можна не лише провести перевірку результатів декодування, а і оцінити пропускну здатність системи, що тестується.

У свою чергу, відлагоджувальна плата дозволяє працювати з модулем ПЛІС сімейства Cyclone V. Дана мікросхема, окрім частини програмної логіки містить апаратні ядра мікропроцесору ARM (доступні два ядра). Завдяки цьому можна об'єднати обчислювальні потужності програмної логіки з гнучкістю налаштувань процесора. На мікропроцесорній частині пристрою можливий, навіть, запуск операційної системи Linux. Дана плата містить достатню кількість логічних ресурсів для проведення тестування роботи декодерів деяких кодів з обраного набору. Для того, щоб мати можливість тестувати декодер при роботі з більшою довжиною кодового слова, необхідно використовувати мікросхему, яка надає більшу кількість ресурсів програмної логіки (наприклад, AlteraStratix та ін.).

Зовнішній вигляд розробленого лабораторного стенду для декодеру TP-кодів на базі ПЛІС представлено на рисунку 2.



Рис. 2. Зовнішній вигляд лабораторного стенду для проведення тестування

Підключення двох модулів організоване наступним чином. На відлагоджувальній платі наявний роз'єм, що відповідає за розташування пінам відомої плати ArduinoUno. У цей роз'єм вставляється перехідна макетна плата, на якій розміщений інтерфейсний модуль. Макетна плата забезпечує з'єднання між шиною даних і допоміжними сигналами на платі та модулем. Такий варіант з'єднання забезпечує надійну передачу сигналів між модулями при використанні високої тактової частоти.

Програмна складова для проведення тестування представляє собою скрипт на мові програмування Python, який окрім безпосередньо декодування вхідних даних, які попередньо були згенеровані та записані у текстовий файл. У текстовому файлі послідовно записана відповідна кількість пакетів. Даний файл використовується як для проведення тестування за допомогою скрипту, так і за допомогою спеціалізованих засобів моделювання (у даному випадку – MentorGraphicsModelSim 10.1) [4]. Як вже зазначалось засоби моделювання демонструють меншу швидкість при моделюванні декодування великої кількості пакетів, тому вони використовуються на початковому етапі налагодження скрипту. Окрім цього, засоби моделювання дозволяють більш детально дослідити роботу кожного окремого модуля, який є складовою частиною декодера. Як результат цього етапу за допомогою розробленого тестового оточення отримані вихідні файли декодованих пакетів, які порівнюються з результатами, які показує скрипт.

При моделюванні роботи декодера використовувався розроблений опис декодера мовою схемотехнічного опису VHDL. Саме цей опис на наступних етапах перенесений на мікросхему ПЛІС для апаратного тестування роботи декодера. Важливим є те, що основні модулі, які переносяться мають бути однаковими як для засобів моделювання, так і для апаратної реалізації на базі ПЛІС.

Моделювання з використанням стандартних програмних засобів, які постачаються виробниками ПЛІС проведено у середовищі ModelSim. При моделюванні на основі розробленого тестового оточення відбувається проходження усіх етапів роботи декодера (завантаження конфігураційних даних, завантаження вхідних даних для декодування, проходження необхідної кількості ітерацій декодування, видача декодованого повідомлення). Скріншот візуального представлення процесу декодування (значення основних сигналів) представлено на рисунку 3.

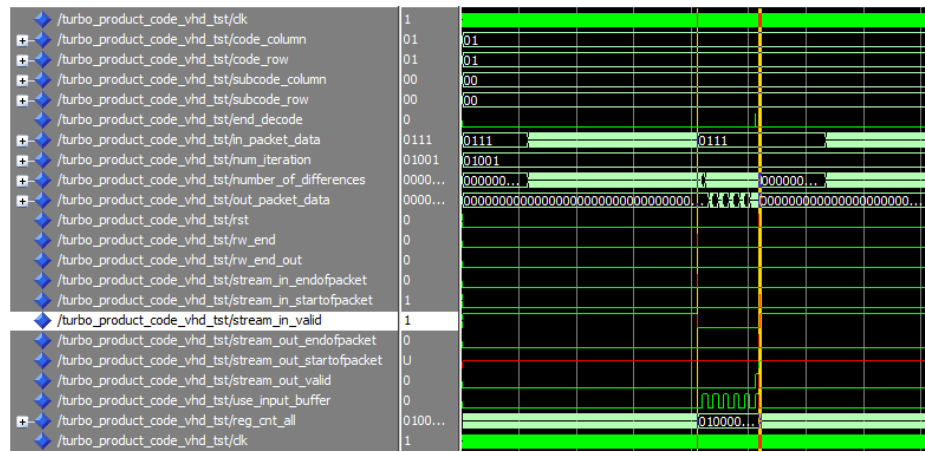


Рис. 3. Скріншот моделювання декодування одного повідомлення розробленим декодером у середовищі ModelSim

На рисунку представлено те, як проходить одна ітерація. Зрозуміло, що найбільше часу у даному випадку займає процес запису повідомлення (перше повідомлення – подальші повідомлення записуються з використанням техніки подвійного буферу), безпосередньо саме декодування (відбулось шість ітерацій декодування) займають набагато менше часу у порівнянні з даним процесом.

Проведено тестування для різних імовірностей помилки для 100 пакетів. Для тестування був обраний код-добутків (64, 57)x(46, 39). Тестування проводилось з використанням розробленого скрипту на мові програмування Python [7]. Для представлення результатів використовувалось програмне середовище gnuplot 5.1 [8]. Результуюча гістограма представлена на рисунку 4.

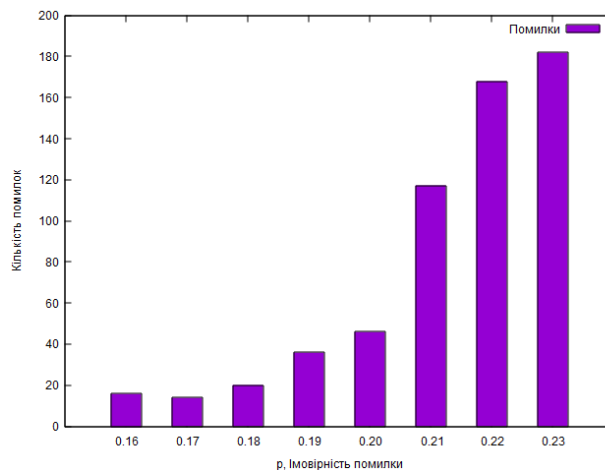


Рис. 4. Гістограма залежності кількості помилок від імовірності помилки для тестових випадків для коду 64*46

Як видно з рисунку, кількість помилок при низькій імовірності є практично однаковою, тому можна зробити висновок, що результативність декодування залежить від характеру помилок, які утворюються.

Таке саме дослідження проведено для коду 32*22. Результати представлені на рисунку 5.

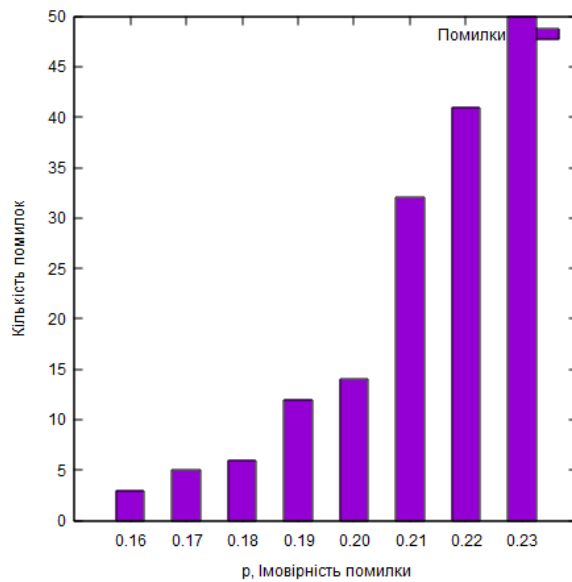


Рис. 5. Гістограма залежності кількості помилок від імовірності помилки для тестових випадків для коду 32*22

У результаті проведення дослідження виявлено, що запропонований метод декодування дозволяє отримати вищу завадостійкість у порівнянні з жорстким декодером (рис. 6а). Крім того, виявлені деякі особливості в роботі декодера, які не дозволяють підвищити продуктивність декодера (рис. 6б).

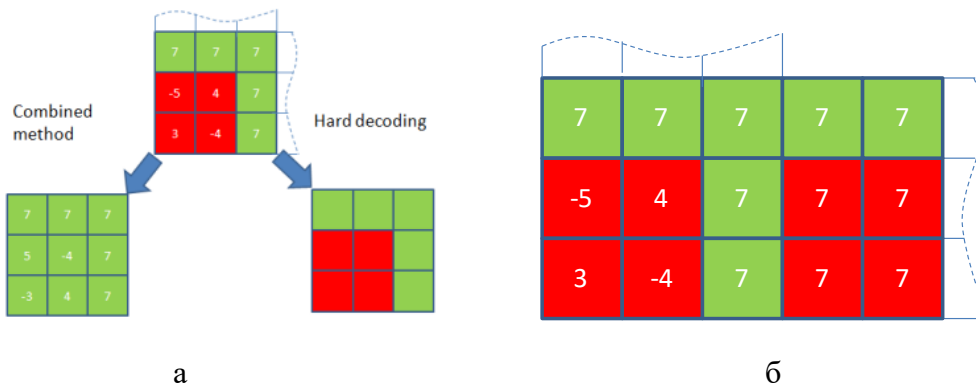


Рис. 6. Типові особливості обробки шаблонів помилок при використанні запропонованого методу декодування: а – порівняння роботи методу; б – шаблон помилок, який не виправляється

Запропонований декодер здатен виправити квадратний шаблон помилок, у той час як жорсткий декодер не виправляє такі помилки. Утім, більш складний шаблон помилок з парою рядків або стовпців з чотирма некоректними значеннями у одних і тих самих позиціях у випадку, якщо він розпізнаний як коректне кодове слово виправлений бути не може. За рахунок зменшення абсолютних значень модулю при декодуванні протилежним декодером вдається виділити біти, які є більш імовірними помилковими бітами. Особливості роботи декодера полягають у тому, що запропонований метод не завжди обирає коректний шлях декодування виділених бітів. Це призводить до ситуації, коли декодер показує відсутність помилок у вихідному слові, в той час як порівняння з початковим кодовим словом вказує на наявність помилок. Дану обставину необхідно враховувати, але оскільки ТР-коди застосовуються в тому числі в якості

складових кодів, то такі помилки можуть бути виправлені декодерами на наступних рівнях.

Висновки

У даній роботі представлена методика проведення тестування декодеру ТР-кодів з використанням програмних та апаратних засобів. Відповідно до представленої методики необхідно забезпечити послідовне виконання кількох стадій для того, щоб мати можливість оцінити виправну здатність декодеру при роботі на заданому коді та пропускну здатність, яка гарантується при заданих умовах. Тестування проводилось з використанням як готових засобів (ModelSim) на основі розробленого тестового оточення, так і з використанням розробленого програмного забезпечення на мові Python. Відносно апаратної частини дослідження, то був розроблений лабораторний стенд для проведення тестування на основі двох відлагоджувальних плат: Altera DE0-SoC-Nanota FTDIFT2232H MiniModule. Поєднання даних двох модулів забезпечує розгортання розробленого схемотехнічного опису мовою VHDL на мікросхемі ПЛІС та дозволяє проводити тестування з пропускну здатністю, яка близька до максимальної для розробленого декодеру. Відповідно до запропонованої методики проведено дослідження роботи декодеру при роботі з різними кодами та за різних умов зашумлення. Дослідження показало, що для деяких видів помилок декодер не здатен забезпечити гарантоване виправлення, тому рекомендованим є використання декількох кодів на рівні з запропонованими для уникнення вказаних типів помилок.

Список літератури

1. Krainyk, Y.M. Low-Complexity High-Speed Soft-Hard Decoding for Turbo-Product Codes. Electronics and Nanotechnologies-2017 / Y.M. Krainyk, V.O. Perov, M.P. Musiyenko. – ELNANO-2017, Kyiv, 2017. – Pp. 471-474.
2. Krainyk, Y.M. Hardware-Oriented Turbo-Product Codes Decoder Architecture / Y.M. Krainyk, V.O. Perov, M.P. Musiyenko, Y. Davydenk. – IDAACS-2017, Bucharest, 2017. – Pp. 151-154.
3. Berrou C. Near Shannon limit error-correcting coding and decoding : Turbo-codes / C. Berrou, A. Glavieux, P. Thitimajshima // IEEE ICC'93, 1993. – Pp. 1064–1071.
4. Download Center. Mode of access: http://fpgasoftware.intel.com/13.1/?product=modelsim_ae#tabs-2 (Date of treatment 27.08.2018).
5. FT2232H Mini Module USB Hi-Speed FT2232H Evaluation Module Datasheet. Mode of access: http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_FT2232H_Mini_Module.pdf (Date of treatment 27.08.2018).
6. DE0-Nano-SoC Kit/Atlas-SoC Kit: Mode of access: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=941> (Date of treatment 27.08.2018).
7. Python. Mode of access: <https://www.python.org/downloads/> (Date of treatment 27.08.2018).
8. Gnuplot. Mode of access: <http://gnuplot.info/> (Date of treatment 27.08.2018).

SOFTWARE AND HARDWARE TESTING OF TURBO-PRODUCT-CODES DECODER

Y.M. Krainyk, V.O. Perov

Petro Mohyla Black Sea National University,
68 Desantnykiv, 10, Mykolaiv, 54003, Ukraine; e-mail: codebreaker7@ukr.net

In this paper we investigate methodic of testing Turbo-Product-codes decoder on software and hardware level. The proposed methodic allows achieving results that characterize decoder's correction ability and through put. The objective of this in vestigation is development of unified methodic that makes it possible to hold testing and evaluation of decoder's properties for any sub-codes of Turbo-Product-code sand for any hard ware platform. The research has allowed emphasizing problems during decoding stage for the

specified algorithm usage and identifying condition for stable work of the decoder. From the practical point of view, it makes possible to define the code that is appropriate for the certain circumstances according to the necessary values of throughput and correction ability. Thus, the decoder can be dynamically configured in obedience to the outer conditions and reach maximum efficient ratio of the two mentioned parameters. The testing methodic supposes division on several stages of testing. In this work several software means have been used during the test stages and among them there is software modeling environment ModelSim, programming language Python. The developed hardware is intended for testing wide range of codes that are supported by the decoder. Hardware tools have also been occupied extensively. Hardware tools include developed laboratory stand using two development boards with FPGA integrated circuit and interface adapter for data transfer between computer and stand. The first one is used for direct implementation of decoder in hardware, while the second one provides necessary throughput capability for transmission speed testing. Combination of developed software and laboratory stand has been utilized during testing in accordance with methodic presented in this work. In compliance with the developed methodic, it is possible to develop universal approach for testing of hardware solutions for error-correcting codes.

Keywords: decoder, Turbo-Product-codes, software testing, hardware testing.

ПРОГРАММНОЕ И АППАРАТНОЕ ТЕСТИРОВАНИЕ TURBO-PRODUCT-КОДОВ

Я.М. Крайник, В.О. Перов

Черноморский национальный университет имени Петра Могилы,
68 Десантников, 10, Николаев, 54003, Украина; e-mail: codebreaker7@ukr.net

В данной работе представлена методика тестирования декодера Turbo-Product-кодов на программном и аппаратном уровнях. Данная методика позволяет получить результаты относительно корректирующей способности декодера, а также пропускной способности. Целью данного исследования является разработка унифицированной методики, которая позволит проводить тестирование и предоставлять оценку показателей декодеров для любых подвидов Turbo-Product-кодов на любой аппаратной платформе. Проведенное исследование позволило определить проблемные места при декодировании при помощи заданного алгоритма и установить, при каких уровнях помех декодер будет работать стабильно. С практической точки зрения это позволяет определить, какой код целесообразно использовать при заданных условиях в соответствии с параметрами пропускной способности и корректирующей способности. Это позволяет проводить конфигурацию декодера динамически в соответствии с изменениями внешних условий для получения максимально эффективного соотношения двух упомянутых параметров. Методика тестирования предполагает прохождение нескольких стадий тестирования. В данной работе использовались программные средства для моделирования ModelSim, язык программирования Python, а также аппаратные средства, среди которых следует выделить разработанный лабораторный стенд на базе двух отладочных модулей с микросхемой FPGA, а также интерфейсной микросхемой для передачи данных между компьютером и стендом. Первая плата используется для непосредственной реализации функций декодера, в то время как вторая способна обеспечить пропускную способность обмена на уровне максимальной пропускной способности системы. Комбинация из разработанного программного обеспечения и лабораторного стенда использовалась при проведении тестирования по представленной в данной работе методикой. В соответствии с разработанной методикой в дальнейшем возможно разработать универсальный подход для тестирования аппаратных решений для помехоустойчивых кодов.

Ключевые слова: декодер, Turbo-Product-коды, программное тестирование, аппаратное тестирование.