

ОСОБЛИВОСТІ ПОБУДОВИ БЛОКІВ ДЕКОДУВАННЯ ДЛЯ ТРС-ДЕКОДЕРУ

Крайник Ярослав, Перов Владислав

Чорноморський національний університет імені Петра Могили

Анотація

У даній роботі представлені особливості побудови деяких блоків реконфігурованого декодера Turbo-Product-кодів. Досліджені та розроблені блоки дозволяють організувати обробку кодових слів для різних кодів, тому декодер, що використовує такі блоки може виконувати декодування різних кодів без необхідності перепрограмування.

Abstract

This paper presents peculiarities of construction of decoding blocks for reconfigurable decoder of Turbo-Product codes. The main attention is devoted to the block of minimum search that is further used for test vector generation. The investigated blocks allow organizing codewords processing for different codes without reprogramming.

Вступ

Декодування Turbo-Product-кодів (англ. TP-codes – ТРС) є важливою складовою побудови інформаційних систем, що базуються на різних стандартах передачі даних (Wi-Fi, DVB-S2 та ін.). Такі системи можуть використовувати програмні логічні інтегральні схеми (ПЛІС) для реалізації основних функцій по декодуванню. Реконфігуровний декодер на базі ПЛІС повинен проводити обробку кодів різної довжини без необхідності додаткового перепрограмування. Це дозволяє змінювати налаштування без необхідності проведення складних маніпуляцій, пов'язаних з даним процесом, завдяки чому сигнал може оброблюватись постійно, і можливим є збільшення пропускну здатності при використанні динамічних протоколів взаємодії. Саме тому проблема побудови реконфігурованого декодера є актуальною.

Декодування ТР-кодів часто базується на алгоритмі Чейза [1] та використовує деякі стадії, що запропоновані у цьому алгоритмі. Важливим моментом у побудові декодерів з використанням даного алгоритму є пошук заданої кількості позицій, що містять мінімальні абсолютні значення у кодовому слові.

Основна частина

У роботі [2] представлена структура для пошуку трьох мінімальних значень у векторів кодового слова для коду, що має довжину 32 біти. Така структура використовується лише для декодування одного коду – (32, 26). У той час, як подача значень на інші коди або неможлива через недостатню кількість входів, або через може призвести до некоректних результатів.

Дана робота є продовження робіт [3, 4], у яких представлений повний алгоритм декодування ТР-кодів, що використовує стадії алгоритму Чейза, а також представлена архітектура декодера на базі ПЛІС, яка враховує апаратні особливості цільової платформи для реалізації.

Для пошуку одного мінімуму може використовуватись каскад, який складається з блоків, представлених на рис. 1. Такий блок надає можливість у кінцевому результаті отримати одне значення та позицію, які дійсно є мінімальним (гарантовано не є більшими за інші значення у кодовому слові). Проте, у ході декодування ТР-кодів необхідно знайти декілька мінімальних значень. Тому необхідною є організація кількох паралельних каскадів.

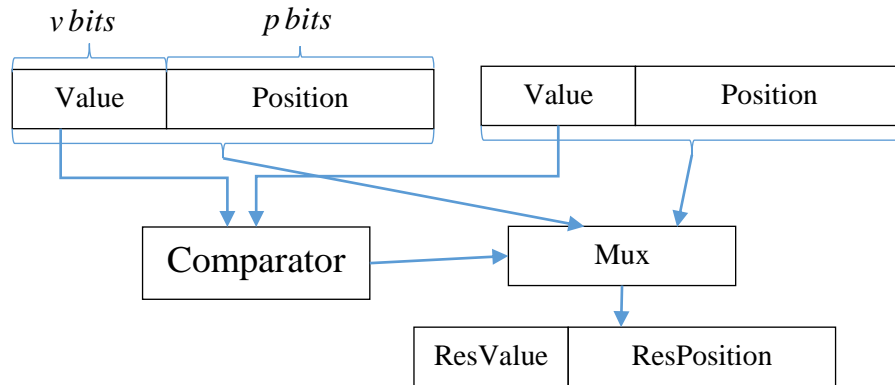


Рисунок 1 – Простий блок для організації каскаду пошуку одиничного мінімуму

Особливістю каскаду блоків порівняння є те, що значення, яке виявилось меншим з двох не відкидається одразу, а подається на вхід паралельного каскаду. Це забезпечує можливість, що обидва вхідні значення, подані на вхід такого елемента, у результаті можуть бути обрані в якості кінцевих мінімумів. Якби одне зі значень відкидалось одразу, то, за умови порівняння двох мінімумів, один з них був би втрачений і позиція мінімуму була б обчислена некоректно. Організація паралельного каскаду надає можливість відкинутому значенню пройти по паралельній гілці та потрапити до числа мінімумів. Значення, які подаються на вхід блоку є комбінацією з самого значення та його індексу. Під значення відводиться фіксована довжина, яка дорівнює v бітам, так само, як і під значення індексу, довжина якого має складати p біт. Для роботи такого каскаду необхідним є інший тип блоків, наведений на рис. 2.

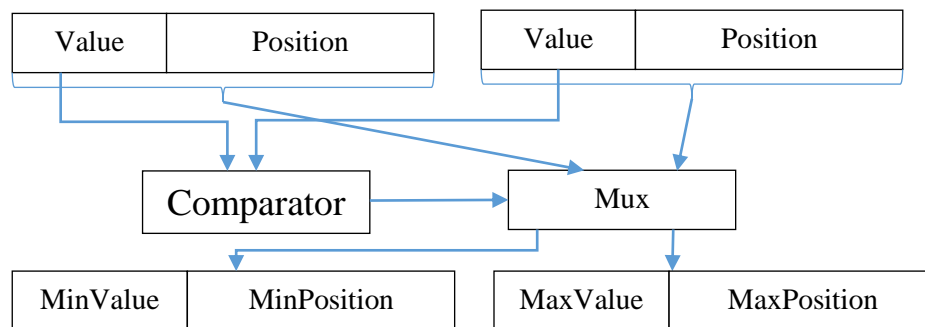


Рисунок 2 – Блок порівняння з двома виходами для організації паралельного каскаду

На відміну від попереднього блоку у даному випадку мультиплексор має два виходи. Один з них призначений для того, щоб обрати мінімальне значення. У випадку рівності значень перевага надається значенню з меншим індексом. На інший вихід подається значення, яке не було обране в якості мінімального.

На обох попередніх рисунках блоки представлені у вигляді регістрів. Також їх можна представити як блоки з відповідною кількістю входів та виходів, оскільки регістри можна розглядати як окремі компоненти, які використовуються для зв'язування блоків комбінаційної логіки.

Іншим важливим моментом у роботі блоку пошуку мінімумів є можливість обробки вхідних даних з різною довжиною кодового слова. Для цього на вхідні регістри, індекси яких перевищують довжину кодового слова, подається значення, яке відображає

максимальне абсолютне значення у системі. Також необхідно забезпечити, щоб усі результуючі значення позицій мінімумів входили до діапазону, що визначається довжиною коду – $[0, n - 1]$.

У випадку, коли значення поза діапазоном не були відсіяні (це можливо, наприклад, коли усі вхідні значення однакові або для деяких варіантів кодових слів), перевага має надаватись значенням з меншим індексом.

Послідовне з'єднання вищенаведених блоків у каскад з кількома рівнями дозволить однозначно визначити мінімальний елемент з найменшим індексом (за умови рівності значень). Проте, для роботи алгоритму на основі генерації списку тестових векторів одного значення недостатньо, тому, такі рівні можуть повторюватись лише до певного моменту, коли буде виділена певна кількість значень, серед яких і необхідно обрати кінцевий набір значень. Для цього, по аналогії з попередньою стадією, коли був визначений один мінімум, організуються блоки, які отримують на вхід результат блоків, які виконують пошук одиничного мінімуму у заданому наборі значень. У ході обробки використовуються обидва типи блоків.

У випадку, коли значення поза діапазоном не були відсіяні (це можливо, наприклад, коли усі вхідні значення однакові або для деяких варіантів кодових слів), перевага має надаватись значенням з меншим індексом.

У даній роботі описані особливості побудови блоків декодування ТР-декодеру, зокрема, блоків пошуку мінімумів. Розроблений блок призначений для використання у реконфігурованому декодері, який здатен проводити обробку різних кодів без необхідності перепрограмування пристрою (цільової платформи для реалізації є ПЛІС). Основна ідея його реалізації полягає у організації паралельного каскаду, що дозволяє значенням, які були відкинуті на одній з початкових стадій, потрапити до заданої кількості мінімальних значень. У каскаді використовуються два типи блоків, коректне з'єднання яких і забезпечує необхідну функціональність. На основі запропонованих у даній роботі положень реалізований декодер, який дозволяє проводити обробку кодових слів довжиною до 128 біт включно. Даний декодер реалізований на базі ПЛІС фірми Altera, яка надана для проведення тестування фірмою «Віаком» (м. Київ) мовою схемотехнічного опису VHDL.

Список використаних джерел:

1. Chase D. "A class of algorithms for decoding block codes with channel measurement information," IEEE Trans. Inform. Theory, vol IT-18, pp. 170–182, Jan. 1972.
2. Leroux C. Turbo-Product code decoder without interleaving resource: From parallelism exploration to high efficiency architecture / C. Leroux, C. Jego, P. Adde, D. Gupta, M. Jezequel / Journal of Signal Processing Systems, Springer. – 2011, 64 (1). – pp.17-29.
3. Krainyk Y. Low-complexity high-speed soft-hard decoding for turbo-product codes / Y. Krainyk, V. Perov, M. Musiyenko – 2017 IEEE 37th International Conference on Electronics and Nanotechnology (ELNANO), 2017. – Kyiv, 2017. - pp. 471-474.
4. Krainyk Y. Hardware-Oriented Turbo-Product Codes Decoder Architecture / Krainyk Y., Perov V., Musiyenko M., Davydenko Ye. // Conference Proceedings of IEEE IDAACS-2017. - Bucharest, 2017. -pp. 151-154.