

На правах рукописи



Мыцко Евгений Алексеевич

**Алгоритмы и аппаратная реализация на ПЛИС устройств
обнаружения и исправления пакетных или независимых ошибок
для сообщений короткой длины**

05.13.05 – «Элементы и устройства вычислительной техники
и систем управления»

Автореферат
диссертации на соискание ученой степени
кандидата технических наук

Томск – 2019

Работа выполнена в федеральном государственном автономном образовательном учреждении высшего образования «Национальный исследовательский Томский политехнический университет» (ФГАОУ ВО НИ ТПУ).

Научный руководитель – доктор технических наук, доцент
Ким Валерий Львович

Официальные оппоненты: **Башкиров Алексей Викторович,**
доктор технических наук, доцент,
исполняющий обязанности заведующего
кафедрой конструирования и производства
радиоаппаратуры Воронежского
государственного технического университета

Останин Сергей Александрович,
кандидат технических наук, доцент,
заведующий кафедрой компьютерной
безопасности Национального
исследовательского Томского
государственного университета

Ведущая организация – Федеральное государственное бюджетное
образовательное учреждение высшего
образования «Ульяновский государственный
технический университет»

Защита диссертации состоится «10» октября 2019 г. в 15 часов 15 минут на заседании диссертационного совета Д 212.268.03, созданного на базе Томского государственного университета систем управления и радиоэлектроники (ТУСУР) по адресу: 634050, г. Томск, ул. Ленина, 40, ауд. 201.

С диссертацией можно ознакомиться в библиотеке ТУСУРа по адресу:
г. Томск, ул. Красноармейская, 146, а также на официальном сайте ТУСУРа:
<https://postgraduate.tusur.ru/urls/vir34nel>

Автореферат разослан «___» _____ 2019 г.

Ученый секретарь
диссертационного совета

Газизов Тальгат Рашитович

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность работы

В настоящее время в эпоху цифровой передачи данных актуальным является обеспечение целостности передаваемых данных от одного вычислительного устройства (ВУ) к другому. Каналы передачи данных могут быть ненадежными в виду воздействия различных шумов и наводок. В свою очередь, в вычислительных устройствах при передаче информации от одного блока к другому любая одиночная ошибка может существенно нарушить процесс вычислений или привести к снижению производительности ВУ. Искажения в данных или командах, возникающие в процессе передачи между элементами вычислительной техники, также обуславливаются старением элементов, ухудшением качества электрических соединений и нестабильностью питания. Из-за отказа отдельных интегральных схем ошибки возникают и при хранении данных, поэтому большая часть неисправностей, как правило, приходится на память. Таким образом, при проектировании устройств передачи (приёма), а также хранения информации требуется реализация способов и алгоритмов обнаружения и исправления ошибок, возникающих в результате воздействия помех или сбоев оборудования.

Одним из известных подходов к обнаружению и исправлению ошибок является применение помехоустойчивого кодирования (ПК) с избыточностью кода.

Первые классы кодов, исправляющие одиночные ошибки, были введены Р. Хэммингом в 1956 году. В 60-х годах XX века Р. Боуз, Д. Рой-Чоудхури, А. Хоквингем (коды Боуза-Чоудхури Хоквингема, БЧХ), И. Рид и Г. Соломон (коды Рида-Соломона, РС) предложили более сильные коды, исправляющие многократные ошибки. Такие коды используются в системах управления, связи и устройствах вычислительной техники по сей день. Из российских ученых большой вклад в исследование помехоустойчивых кодов внесли Харкевич А.А., Колесников В.Д., Золотарёв В.В., Зяблов В.В. В процессе развития вычислительной техники, предложенные алгоритмы исправления ошибок нашли применение в различных устройствах передачи и хранения данных.

После открытия мощных кодов, исправляющих многократные ошибки, актуальной стала задача разработки эффективных способов и алгоритмов их построения и декодирования. Первые алгоритмы для построения и декодирования кодов БЧХ и Рида-Соломона с практическим применением были предложены У. Питерсоном и Э. Берлекэмпом. В настоящее время исследованию способов декодирования различных помехоустойчивых кодов посвящены работы российских ученых: Золотарева В.В. (многопороговые декодеры), Гладких А.А. (декодирование блоковых и избыточных кодов), Егорова С.И. (декодеры низкоплотностных кодов и кодов Рида-Соломона), Башкирова А.В. (реализация многофункциональных декодеров на программируемых логических интегральных схемах (ПЛИС)).

В 1961 году У. Питерсоном был предложен отдельный вид помехоустойчивых кодов, позволяющих обнаруживать ошибки без прямого исправления – циклический избыточный код CRC (Cyclic redundancy code). Основной целью кодов CRC было эффективное обнаружение ошибок с применением наименьшей и фиксированной избыточности. Исправление ошибок на основе таких кодов осуществляется за счёт повторной передачи пакетов данных.

На данный момент CRC-коды длиной от 4 до 64 бит (CRC4, CRC8, CRC16, CRC32, CRC64) используются в протоколах передачи данных ETHERNET, ZigBee, ModBus, DLMS/SLIP, а также в архиваторах WinRAR, WinZIP. В сетевых платах ЭВМ проверка целостности данных с применением CRC-кодов аппаратно реализуются в составе модуля FCS (Frame check sequence). Помехоустойчивые коды, исправляющие ошибки, широко используются в стандартах цифрового телевизионного вещания DVB-T2, системах спутниковой, сотовой связи, системах автоматизации производства и телекоммуникациях, системах дистанционного зондирования Земли. Применение помехоустойчивых кодов, обнаруживающих и исправляющих ошибки, является одним из способов повышения надежности цифровых и запоминающих устройств ЭВМ. Современные модули памяти SDRAM DDR реализуются с аппаратной поддержкой алгоритмов исправления ошибок на основе кодов Хэмминга и дополнительных модулей хранения кодов ECC (Error-correcting code).

Существующие алгоритмы обнаружения ошибок, применяемые в устройствах с контролем целостности данных и основанные на применении циклических избыточных кодов, позволяют эффективно обнаруживать ошибки в больших пакетах данных, однако либо обладают низким быстродействием (классический побитовый алгоритм), либо при модификации требуют достаточно большой объем памяти (табличный алгоритм). Таким образом, актуальной является задача разработки алгоритма вычисления циклических избыточных кодов, который будет компромиссным вариантом по быстродействию и аппаратным затратам при реализации на микроконтроллерах и ПЛИС.

Циклические помехоустойчивые коды (ЦПК) БЧХ, исправляющие независимые ошибки, обычно записываются в формате (n, m, d) , где n – длина кодового слова, m – длина информационного блока, d – расстояние Хэмминга. Например, ЦПК БЧХ $(15, 7, 5)$ имеет длину кодового слова 15 бит, длину информационного блока 7 бит и расстояние Хэмминга, равное 5. Длины кодовых слов таких кодов ограничены выражением $n = 2^h - 1$, где h – натуральное число. Данный факт не позволяет построить помехоустойчивый код для любых n и m без применения операции укорачивания. Это приводит к снижению эффективности кода, а также к низкой эффективности использования аппаратных ресурсов при разработке устройств исправления ошибок. Коды Рида-Соломона, являющиеся частным случаем кодов БЧХ, имеют меньшую избыточность, однако ориентированы на исправление только

пакетных ошибок. Для реализации методов и алгоритмов декодирования помехоустойчивых кодов БЧХ и Рида-Соломона требуется знание арифметики полей Галуа и решение ключевых уравнений, что увеличивает сложность разработки устройств исправления ошибок. Таким образом, существует проблема построения помехоустойчивых кодов, аналогичных кодам БЧХ и Рида-Соломона, но более эффективных по избыточности и для любых длин информационного сообщения без применения операции укорачивания.

Во многих системах с защитой передаваемых данных от ошибок модули обнаружения и исправления ошибок являются служебными (вспомогательными), а основные ресурсы памяти и вычислительные мощности устройств используются для решения задач согласно прямому назначению системы. В связи с этим, сформулируем следующие требования к способам, алгоритмам и устройствам, обеспечивающим целостность информации:

- высокая эффективность – наилучшее отношение полезной информации к избыточной;
- простота реализации – низкая сложность алгоритмов, а также наименьшая область знаний для программной и аппаратной реализаций;
- быстродействие – минимальное время (количество тактов устройства) для обнаружения и исправления ошибок;
- минимальные аппаратные затраты – наименьшее количество логических элементов модуля исправления ошибок и небольшой требуемый объем памяти запоминающего устройства для реализации в системах с дефицитом ресурсов.

Таким образом, актуальными задачами являются разработка программ и аппаратная реализация устройств обнаружения и исправления ошибок, удовлетворяющих сформулированным ранее требованиям с применением новых и модификацией существующих алгоритмов. С учетом описанных выше проблем и задач сформулируем цель настоящей работы.

Целью работы является разработка эффективных алгоритмов, программ и устройств обнаружения и исправления пакетных или независимых ошибок при передаче и хранении данных с информационным сообщением короткой длины.

Для выполнения поставленной цели необходимо выполнить следующие задачи:

1. Провести обзор и анализ существующих способов и алгоритмов обнаружения и исправления ошибок, применяемых в устройствах передачи и хранения данных.
2. Разработать алгоритм быстрого вычисления циклических избыточных кодов CRC для реализации на микроконтроллерах и ПЛИС. Реализовать разработанный алгоритм на микроконтроллере и ПЛИС для сравнения с существующими алгоритмами.
3. Разработать алгоритм поиска образующих полиномов, позволяющий находить полиномы, применяемые для построения более эффективных циклических помехоустойчивых кодов, чем коды БЧХ.

4. Разработать устройства исправления независимых ошибок на ПЛИС с использованием известных циклических помехоустойчивых кодов короткой длины и предложенного помехоустойчивого кода, построенного на основе результатов поиска образующего полинома, для сравнения быстродействия и аппаратных затрат.
5. Осуществить поиск образующих полиномов для построения циклических кодов, исправляющих пакетные ошибки. Разработать алгоритм и устройства исправления пакетных ошибок на ПЛИС на основе циклического помехоустойчивого кода, исправляющего пакетные ошибки.

Объектом исследования являются программы и устройства на ПЛИС для обнаружения и исправления пакетных или независимых ошибок.

Предметом исследования являются алгоритмы обнаружения и исправления пакетных или независимых ошибок для реализации на ПЛИС.

Методы исследований. Для достижения поставленной цели в работе применены методы компьютерного моделирования, разработки программ на языках программирования низкого и высокого уровней; основы цифровой схемотехники и разработки цифровых устройств с применением систем автоматизированного проектирования.

Достоверность и обоснованность предложенных алгоритмов, программ и аппаратных реализаций подтверждается результатами экспериментальных исследований, компьютерным моделированием в САПР Quartus II и публикациями основных результатов работы в ведущих российских изданиях и трудах зарубежных конференций.

Научную новизну полученных в работе результатов определяют:

1. Матричный алгоритм вычисления контрольной суммы CRC, отличающийся от известных лучшим быстродействием при реализации на ПЛИС и меньшим требуемым объёмом памяти при программной реализации.
2. Алгоритм поиска образующих полиномов, адаптированный для параллельных вычислений и отличающийся от известных тем, что позволяет получать полиномы более короткой длины для построения циклических помехоустойчивых кодов, более эффективных с точки зрения отношения полезной информации к избыточной, чем коды BCH.
3. Предложенные быстродействующие декодирующие устройства на основе циклического помехоустойчивого кода (17, 9, 5) с меньшими аппаратными затратами, чем устройства на основе укороченного кода BCH (19, 9, 5).
4. Предложенная модификация циклического алгоритма декодирования с применением образующих полиномов, полученных с помощью программы поиска, позволяющая исправлять пакетные ошибки для циклических помехоустойчивых кодов без ограничения длины кодового слова.

Практическая значимость работы

Практически значимыми являются программы вычисления контрольной суммы CRC с применением предложенного алгоритма, аппаратные реализации устройств вычисления CRC8 и CRC32 на ПЛИС, программы вычисления CRC8

на языке Assembler для микроконтроллера ATtiny44, алгоритмы и программы поиска образующих полиномов на языке C++ с применением технологии OpenMP. Разработанные устройства исправления независимых и пакетных ошибок на ПЛИС с применением табличного и циклического алгоритмов декодирования обеспечивают целостность информации в устройствах вычислительной техники. По результатам исследований получено 4 свидетельства о регистрации программы для ЭВМ.

Работа выполнена в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014—2020 годы», соглашение № 05.578.21.0272 от 20.12.2018 г. (уникальный идентификатор проекта RFMEFI57818X0272) по теме: «Разработка технических решений и аппаратно-программного комплекса управления цифровыми электрическими подстанциями для построения интеллектуальной энергосистемы» и хозяйственного договора № 4-673/16У от 26.09.2016 «Разработка средств технического обеспечения устройств сбора данных, контроля и защиты электрофизической установки токамак» (заказчик ООО «ТомИУС-проект», г. Томск).

Реализация и внедрение результатов работы

Полученные алгоритмы, программы и аппаратные модули на ПЛИС применены в системе управления электрофизической установки Токамак КТМ (Национальный ядерный центр, Республика Казахстан, г. Курчатов), в том числе разработанное устройство на ПЛИС для исправления пакетных ошибок длиной до 3 бит при длине сообщения 8 бит с модулем вычисления CRC8 в системе передачи команд от центрального блока подсистемы синхронизации к локальным модулям синхронизации.

Отдельные результаты диссертационного исследования использованы в учебном процессе НИ ТПУ в дисциплинах «Микропроцессоры и микроконтроллеры», «Программирование на языках описания аппаратуры».

Основные положения, выносимые на защиту

1. Матричный алгоритм вычисления циклических избыточных кодов CRC позволяет при программной реализации задействовать до 32 раз меньший объем памяти для хранения предвычисленных значений, а при аппаратной реализации на ПЛИС увеличить быстродействие устройств вычисления CRC до 8,5 раз по сравнению с табличным алгоритмом.
2. Алгоритм поиска образующих полиномов, адаптированный для параллельных вычислений, позволяет находить полиномы для построения циклических помехоустойчивых кодов, имеющих избыточность до 25 % меньше, чем у циклических кодов BCH с сообщением длиной от 2 до 32 бит и количеством исправляемых ошибок от 2 до 4.
3. Разработанное устройство исправления независимых ошибок на ПЛИС с применением циклического помехоустойчивого кода (17, 9, 5) на основе циклического алгоритма декодирования обладает в 2 раза лучшим быстродействием и позволяет повысить эффективность использования логических элементов в 2,6 раза по сравнению с устройством на основе

укороченного кода БЧХ (19, 9, 5) с аналогичной длиной информационного блока.

4. Разработанное устройство исправления пакетных ошибок на ПЛИС с применением модифицированного циклического алгоритма декодирования и образующих полиномов, полученных с помощью программы поиска, позволяет исправлять пакетные ошибки для циклических помехоустойчивых кодов без ограничения длины кодового слова.

Соответствие результатов паспорту научной специальности

В работе приведены результаты создания принципиально новых схем и устройств с применением языка описания аппаратуры для решения задачи контроля целостности информации, передаваемой между элементами вычислительной техники, что соответствует п.1 паспорта научной специальности 05.13.05 Элементы и устройства вычислительной техники и систем управления. В соответствии с п.4. паспорта научной специальности разработаны алгоритмы и программы контроля целостности данных в устройствах вычислительной техники.

Апробация работы. Основные результаты работы обсуждались на следующих конференциях и семинарах:

Международная научно-практическая конференция «Молодежь и современные информационные технологии» (г. Томск, 2012 – 2017 гг.), Международная научно-практическая конференция студентов, аспирантов и молодых учёных «Современные техника и технологии» (г. Томск, 2012, 2014 гг.), 9-й Международный Форум по Стратегическим Технологиям 2014, IFOST-2014 (г. Дакка, Бангладеш, 2014 г.), XII Международная IEEE Сибирская конференция по управлению и связи, SIBCON-2016 (г. Москва, 2016 г.), III Международная научная конференция «Информационные технологии в науке, управлении, социальной сфере и медицине» (г. Томск, 2016 г.), 55-ой Международная научная студенческая конференция МНСК-2017, (г. Новосибирск, 2017 г), Международная научно-техническая конференция студентов, аспирантов и молодых ученых, посвященная 55-летию ТУСУРа, «Научная сессия ТУСУР-2017» (г. Томск, 2017 г.), III Всероссийский молодежный научный форум «Наука будущего – наука молодых», (г. Нижний новгород, 2017 г), Всероссийская научная конференция молодых ученых «Наука. Технологии. Инновации», (г. Новосибирск, 2017 г), VIII Международная научно-практическая конференция «Высокопроизводительные вычислительные системы и технологии в научных исследованиях, автоматизации управления и производства», ВВСТ– 2018 (г. Барнаул, 2018 г.).

Результаты диссертационной работы были отмечены медалью Российской академии наук в области информатики, вычислительной техники и автоматизации за научно-исследовательскую работу «Исследования матричного алгоритма вычисления контрольной суммы CRC и его аппаратная реализация» по итогам конкурса 2014 г (постановление Президиума РАН № 24 от 17.02.2015 г.).

Публикации

Результаты диссертационного исследования опубликованы в 6 статьях рецензируемых изданий, рекомендованных ВАК РФ. Четыре публикации индексируются в Международной базе данных SCOPUS (Conference paper) и 6 публикаций – в Web of Science (Conference paper). Получено 4 свидетельства о регистрации программы для ЭВМ.

Личный вклад

Основные научные результаты, выносимые на защиту, получены автором самостоятельно. В работах, опубликованных в соавторстве, личный вклад состоит в следующем: в публикациях [1, 2, 5, 11, 12, 17, 18, 22] автором реализован на ПЛИС, микроконтроллере и под ОС Linux матричный алгоритм вычисления контрольной суммы CRC. В работах [3, 7, 8, 15, 16, 19, 25] автором предложены и реализованы алгоритмы поиска образующих полиномов для построения циклических помехоустойчивых кодов, исправляющих независимые и пакетные ошибки. Поставлены компьютерные эксперименты по поиску образующих полиномов с применением технологий параллельных вычислений. В публикациях [4, 9, 10, 19, 23, 24] автором проведены исследования аппаратных реализаций на ПЛИС циклического алгоритма декодирования на примере кодов БЧХ короткой длины (до 15 бит). В работах [21, 22] автором предложены и реализованы устройства исправления ошибок на основе циклического помехоустойчивого кода (17, 9, 5), обладающего лучшими характеристиками по сравнению с кодом БЧХ (15, 7, 5) и укороченным кодом БЧХ (19, 9, 5). Предложенный автором циклический алгоритм декодирования и устройство на ПЛИС для исправления пакетных ошибок представлены в работах [6, 13, 14].

Структура и объём работы. Диссертация состоит из введения, четырех глав, заключения, списка литературы из 121 наименования и пяти приложений. Объём диссертации составляет 217 страниц, включая 102 рисунка и 40 таблиц.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обосновывается актуальность диссертационного исследования, формулируются цель и задачи исследования, а также приводится краткое содержание работы по главам.

Первая глава посвящена анализу современных способов и алгоритмов обнаружения и исправления пакетных и независимых ошибок при передаче и хранении данных.

Рассмотрены известные алгоритмы вычисления циклических избыточных кодов CRC. Анализ показал, что существуют два основных алгоритма вычисления CRC: классический с применением последовательного сдвига бит данных через регистр и операции сложения по модулю 2 (Исключающее ИЛИ), а также табличный, ускоряющий процесс вычисления CRC за счет применения таблицы предвычисленных значений. Недостатком первого алгоритма является низкая скорость вычисления из-за побитового сдвига. Недостатком табличного алгоритма является то, что требуемый объем памяти для хранения таблицы

предвычисленных значений (256 байт для CRC8, 1 Кб для CRC32) превышает объем памяти программ для исходного кода вычисления CRC.

Проанализированы существующие способы и алгоритмы исправления пакетных и независимых ошибок при передаче данных. Обозначена проблема выбора образующего полинома, позволяющего строить эффективные помехоустойчивые коды, исправляющие как независимые, так и пакетные ошибки, а также алгоритма декодирования данных кодов, требующего минимальных аппаратных и вычислительных затрат для его реализации.

Во **второй** главе предложен матричный алгоритм (рис. 1) вычисления циклического избыточного кода CRC с модификациями, позволяющими увеличить размер блока данных, обрабатываемого за итерацию (длина сдвига).

Матрица в предложенном алгоритме вычисляется следующим образом:

$$G = \begin{pmatrix} x^k \bmod G(x) \\ x^{k+1} \bmod G(x) \\ \dots \\ x^{k+m-1} \bmod G(x) \end{pmatrix}$$

где $G(x)$ – образующий полином, k – разрядность CRC, m – длина блока данных, обрабатываемого за одну итерацию алгоритма.

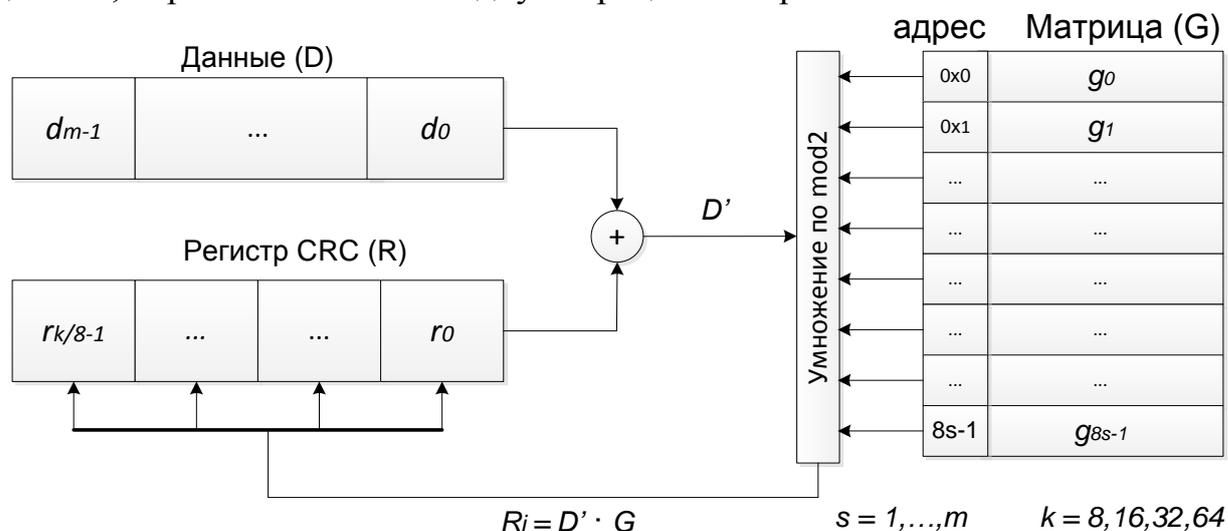


Рисунок 1 – Структурная схема матричного алгоритма CRC

Параметр k указывает разрядность контрольной суммы CRC. Параметр s показывает количество байт, обрабатываемых за итерацию. При $s = 1$ CRC вычисляется побайтно по аналогии с табличным алгоритмом. Контрольная сумма CRC на i -ой итерации получается путём умножения вектора $D' = (d'_0, d'_1, \dots, d'_{m-1}) = R$ хог D на матрицу $G = (g_0, g_1, \dots, g_{8s-1})$.

Установлено, что для реализации матричного алгоритма требуется в 32 раза меньший объем памяти, чем для реализации табличного алгоритма (табл. 1). Для реализации четырехбайтового сдвига требуется объём памяти в 8 раз меньше по сравнению с табличным алгоритмом.

Таблица 1 – Сравнение размеров матрицы и таблицы (в байтах)

Контрольная сумма	Сдвиг (s)					Табличный
	1 байт	2 байта	4 байта	32 байта	1024 байта	
CRC8	8	16	32	256	8192	256
CRC16	16	32	64	512	16384	512
CRC32	32	64	128	1024	32768	1024
CRC64	64	128	256	2048	65536	2048

Поставлен компьютерный эксперимент по вычислению контрольной суммы CRC32 для различных данных размером от 10 до 1010 Мб на суперкомпьютерном кластере «СКИФ-политех1» с процессором Intel XEON 5150, 2.66 ГГц и оперативной памятью объёмом 8 Гб. Для больших файлов CRC необходимо вычислять по частям, помещая блоки файла в буфер заданного размера. В связи с этим, исследование проводилось в несколько этапов: с буферами в 1 Мб, в 4 байта и в 1 байт – для моделирования вычисления контрольной суммы в системах с малым доступным объёмом памяти.

Установлено, что реализация матричного алгоритма с обработкой по 4 байта за итерацию и буфером равным 4 байта, является более быстродействующей, чем реализация табличного алгоритма (рис. 2).

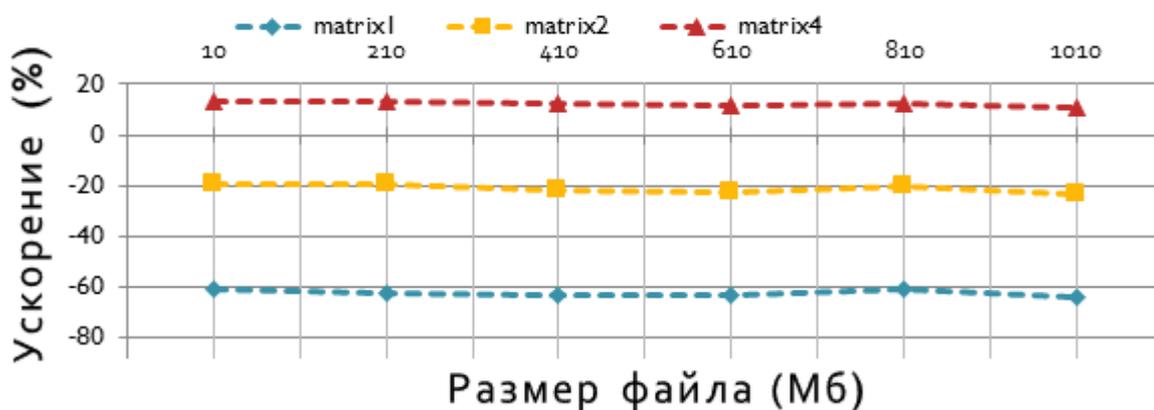


Рисунок 2 – Ускорение (отставание) матричного алгоритма с обработкой по 4 байта относительно табличного алгоритма

Поставлен эксперимент на микроконтроллере Attiny44 по вычислению контрольной суммы CRC8 для системы измерения температуры с применением датчика DS18B20. По результатам эксперимента установлено, что в случаях, когда ресурсы для реализации дополнительных модулей контроля целостности значительно ограничены, целесообразна реализация матричного алгоритма с матрицей в памяти RAM или в регистрах общего назначения.

Поставлен эксперимент по вычислению контрольных сумм CRC8 и CRC32 на ПЛИС Cyclone III от Altera. В таблице 2 приведены результаты эксперимента по вычислению контрольной суммы CRC8 на ПЛИС для пакета данных длиной 8 байт.

Таблица 2 – Характеристики устройств вычисления CRC8

Алгоритм	Макс. частота, МГц	Такты	Мин. время, нс	Ячейки		Память, бит
				модуль приёма	модуль CRC8	
Табличный	182	8	43,9	72	8	2048
Матричный 1 байт	386	8	20,7	72	13	0
Матричный 2 байта	295	4	13,5	80	21	0
Матричный 4 байта	266	2	7,5	96	38	0
Матричный 8 байт	233	1	4,3	65	54	0

Аппаратная реализация однобайтового матричного алгоритма является примерно в 2 раза более быстросействующей, чем реализация табличного. Однако для реализации матричного алгоритма требуется на 38 % больше логических элементов, при этом для реализации таблицы предвычисленных значений (табличного алгоритма) требуется 2048 бит памяти ROM.

Увеличение длины блока, обрабатываемого за итерацию, позволяет увеличить быстросействие устройства вычисления CRC8 за счет уменьшения количества его тактов. При этом максимальная частота работы снижается из-за увеличения критического пути прохождения сигнала. При обработке блока длиной 8 байт за итерацию контрольная сумма CRC8 вычисляется за 1 такт при максимальной частоте 233 МГц, что позволяет достичь прироста производительности до 10 раз. Однако количество логических элементов при этом увеличится в 6,75 раз. В таблице 3 приведены результаты эксперимента по вычислению контрольной суммы CRC32 на ПЛИС для пакета данных объёмом 64 байта.

Таблица 3 – Характеристики устройств вычисления CRC32

Алгоритм	Макс. частота, МГц	Такты, шт	Мин. время, нс	Ячейки, шт		Память, бит
				буфер данных	модуль CRC32	
Табличный (с ROM)	181	64	353,1	520	40	8192
Матричный 1 байт	388	64	164,9	520	48	0
Матричный 2 байта	289	32	110,7	528	89	0
Матричный 4 байта	228	16	70,1	544	155	0
Матричный 8 байт	194	8	41,2	577	251	0
Матричный 64 байта	102	1	9,8	519	1203	0

При побайтовой обработке матричный алгоритм вычисляет CRC32 примерно в 2 раза быстрее, чем табличный. Увеличение длины блока, обрабатываемого за итерацию, несмотря на падение максимальной частоты из-за увеличения критического пути прохождения сигнала, приводит к росту быстродействия за счёт уменьшения количества тактов, необходимых для обработки всего пакета данных.

Таким образом, наиболее эффективной является реализация матричного алгоритма вычисления CRC32 с побайтовой обработкой, так как быстродействие устройства увеличивается примерно в 2 раза, а аппаратные затраты при этом возрастают всего на 20%. Увеличение размера блока, обрабатываемого за итерацию в матричном алгоритме до 8 байт, позволяет повысить быстродействие до 8,5 раз относительно табличного алгоритма при росте количества логических элементов модуля вычисления CRC32 до 6,2 раз. Увеличение размера обрабатываемого блока при матричном алгоритме до 64 байт позволяет повысить быстродействие до 36 раз при росте количества логических элементов до 30 раз.

В **третьей** главе предложен алгоритм поиска образующих полиномов, отличающийся от известных тем, что позволяет находить полиномы для построения циклических помехоустойчивых кодов более эффективных по скорости кода, чем коды БЧХ.

Поставлен компьютерный эксперимент по поиску образующих полиномов с применением технологий параллельных вычислений для построения кодов, исправляющих независимые и пакетные ошибки. Разработанный алгоритм поиска позволяет находить полиномы для построения кодов, длина которых на 1–5 бит короче, чем коды БЧХ при длине информационного блока до 32 бит и кратности исправляемых ошибок до 4. Применение технологий параллельных вычислений позволяет значительно ускорить процесс поиска полинома для большинства из исследуемых длин информационного блока.

Предложено устройство исправления независимых ошибок (рис. 3) на ПЛИС с применением циклического помехоустойчивого кода (17, 9, 5), построенного по результатам работы алгоритма поиска (табл. 4). Код обладает большей скоростью, чем код БЧХ (15, 7, 5), и позволяет кодировать более длинный информационный блок при сохранении избыточности.

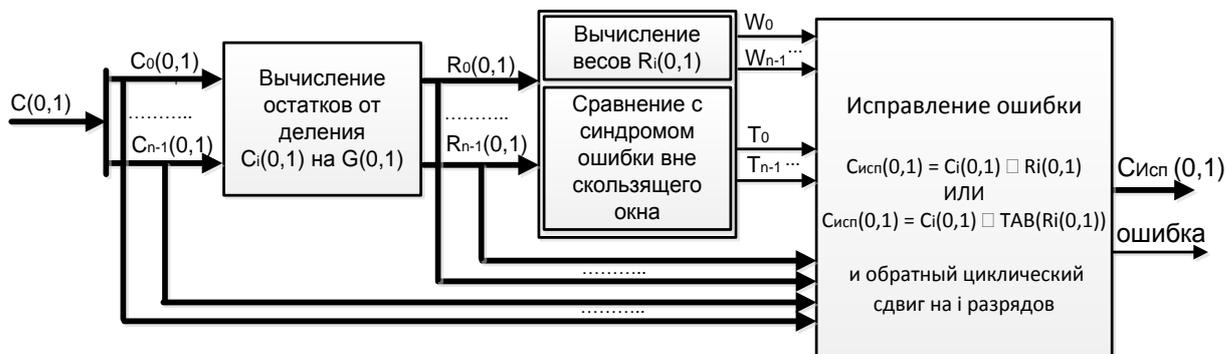


Рисунок 3 – Структурная схема декодера помехоустойчивого кода (17, 9, 5)

Таблица 4 – Длины кодовых слов для образующих полиномов:
 А–коды, построенные по найденному полиному, Б–коды БЧХ,
 В–разность длин А – Б

Код	А			Б			В		
	2	3	4	2	3	4	2	3	4
1	-	-	-	-	-	-	-	-	-
2	8	-	-	10	-	-	-2	-	-
3	11	13	-	11	13	-	0	0	-
4	12	14	19	12	14	24	0	0	-5
5	13	15	21	13	15	25	0	0	-4
6	14	17	26	14	21	26	0	-4	0
7	15	18	27	15	22	27	0	-4	0
8	16	19	28	18	23	28	-2	-4	0
9	17	20	29	19	24	29	-2	-4	0
10	19	21	30	20	25	30	-1	-4	0
11	20	22	31	21	26	31	-1	-4	0
12	21	23	32	22	27	36	-1	-4	-4
13	23	28	33	23	28	37	0	0	-4
14	24	29	34	24	29	38	0	0	-4
15	25	30	35	25	30	39	0	0	-4
16	26	31	36	26	31	40	0	0	-4
17	27	33	37	27	35	41	0	-2	-4
18	28	34	38	28	36	42	0	-2	-4
19	29	36	39	29	37	43	0	-1	-4
20	30	37	40	30	38	44	0	-1	-4
21	31	38	41	31	39	45	0	-1	-4
22	33	39	45	34	40	46	-1	-1	-1
23	35	40	46	35	41	47	0	-1	-1
24	36	41	47	36	42	48	0	-1	-1
25	37	42	49	37	43	49	0	-1	0
26	38	43	50	38	44	50	0	-1	0
27	39	44	51	39	45	51	0	-1	0
28	40	45	52	40	46	52	0	-1	0
29	41	46	53	41	47	53	0	-1	0
30	42	47	54	42	48	54	0	-1	0
31	43	48	55	43	49	55	0	-1	0
32	44	49	56	44	50	56	0	-1	0

На рисунке 4 представлен алгоритм поиска образующих полиномов для построения ЦПК, исправляющих независимые ошибки. Анализ алгоритма показал, что наиболее трудоёмким является этап проверки расстояния Хэмминга для кодовых слов (этап 2), построенных на основе полинома-кандидата. При этом данный этап подлежит распараллеливанию путём распределения кодовых слов по вычислительным ядрам (потокам).

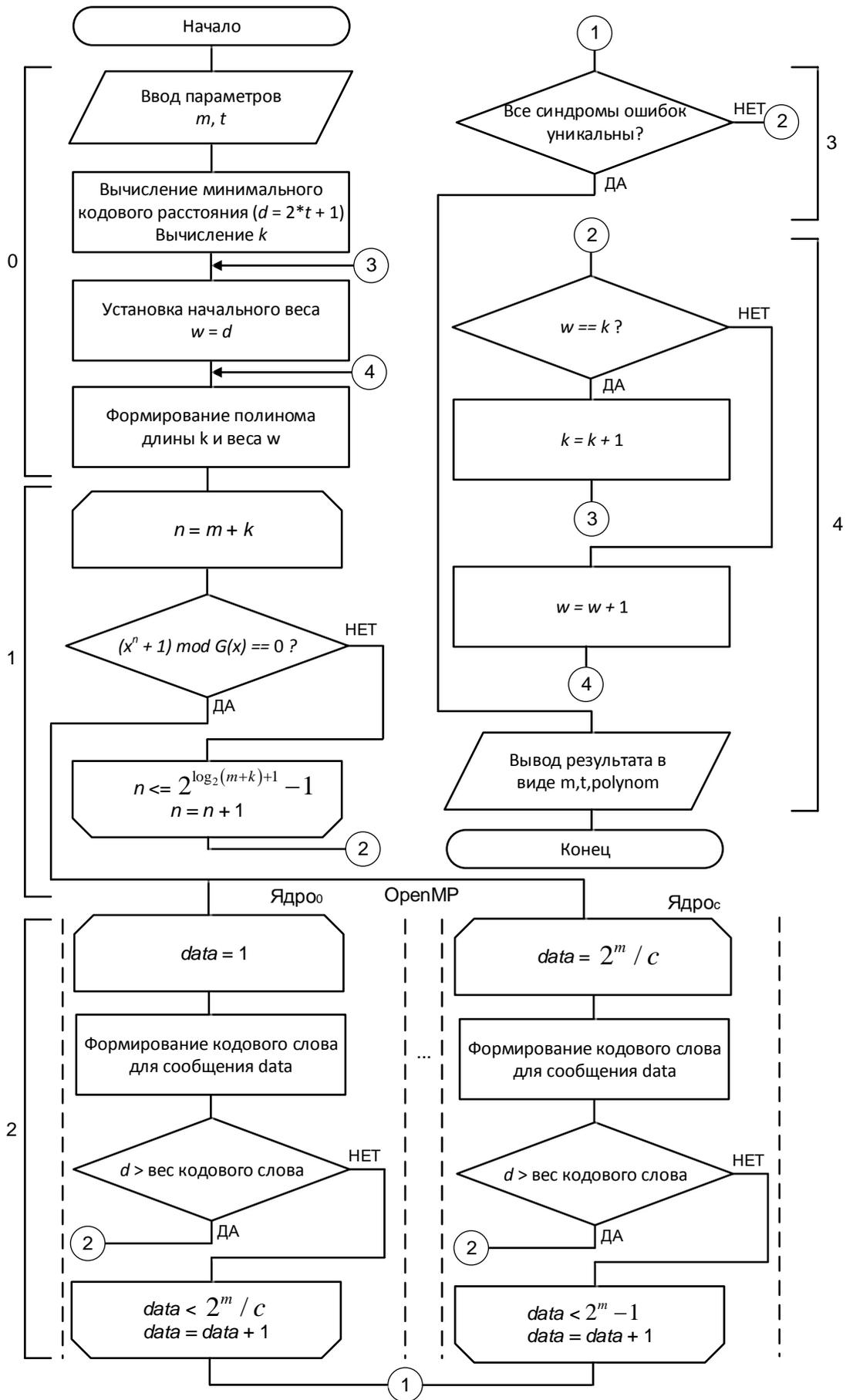


Рисунок 4 – Алгоритм поиска с распараллеленным этапом 2

Проведено сравнение характеристик разработанных устройств исправления ошибок на ПЛИС с применением предложенного помехоустойчивого кода (17, 9, 5) и укороченного кода БЧХ (19, 9, 5). Сделаны выводы о том, что декодирующие устройства с применением кода (17, 9, 5) являются более быстродействующими и менее аппаратно-затратными, чем устройства с применением укороченного кода БЧХ (19, 9, 5) (табл. 5).

Таблица 5 – Характеристики исследуемых устройств декодирования

Алгоритм	Макс. частота, МГц	Такты	Мин. время, нс	Кол-во ячеек	Память ROM, бит
Декодер кода БЧХ (15, 7, 5)					
Табличный	429	3	6,9	19	1792
Циклический (классический)	333	16	48	70	0
Циклический (параллельный)	194	2	10,2	179	0
Декодер кода (17, 9, 5)					
Табличный	350	3	8,4	19	2304
Циклический (классический)	258	18	69,3	103	0
Циклический (параллельный)	177	2	11,2	287	0
Декодер укороченного кода БЧХ (19, 9, 5)					
Табличный	348	3	8,6	21	9216
Циклический (классический)	181	11	60,7	159	0
Циклический (параллельный)	87	2	23	767	0

Предложена модификация циклического алгоритма декодирования, позволяющая исправлять пакетные ошибки, с применением образующих полиномов, найденных с помощью алгоритма поиска для пакетных ошибок. Модифицированный алгоритм можно представить следующим образом:

Начало.

Шаг 1. Вычисляется остаток от деления по модулю 2 кодового слова на образующий полином. Если остаток нулевой, то кодовое слово принято без ошибок и алгоритм завершает работу. Иначе переход на шаг 2.

Шаг 2. Остаток от деления кодового слова на образующий полином проверяется на совпадение с одним из шаблонов ошибки длины p . Если совпадений нет, то кодовое слово циклически сдвигается на 1 разряд вправо и осуществляется переход на шаг 4. Иначе переход на шаг 3.

Шаг 3. Кодовое слово суммируется по модулю 2 с остатком от деления. Результат сдвигается циклически влево на количество разрядов, равное количеству предшествующих сдвигов вправо. Алгоритм завершает работу.

Шаг 4. Если достигли максимального количества сдвигов кодового слова $n - 1$, где n – длина кодового слова, и не нашли подходящего остатка (условия

описаны в шаге 2), то алгоритм завершает работу с флагом обнаружения неисправимой ошибки. Иначе переход на шаг 1.

Конец.

Описаны и исследованы разработанные декодеры циклического помехоустойчивого кода (15, 8, 3) на ПЛИС, исправляющего пакетные ошибки длиной до 3 бит с применением табличного и циклического алгоритмов декодирования. Установлено, что декодеры на основе ЦПК (15, 8, 3) работают на большей частоте и при параллельной реализации циклического алгоритма декодирования требуют меньших аппаратных ресурсов, чем устройства на основе кода БЧХ (15, 7, 5) (табл. 6).

Таблица 6 – Характеристики устройств исправления пакетных ошибок

Алгоритм	Макс. частота, МГц	Такты	Мин. время, нс	Кол-во ячеек	Память ROM, бит
Декодер кода БЧХ (15, 7, 5)					
Табличный	429	3	6,9	19	1792
Циклический (классический)	333	16	48	70	0
Циклический (параллельный)	194	2	10,2	179	0
Декодер кода (15, 8, 3)					
Табличный	565	3	5,4	17	1024
Циклический (классический)	436	10	22,9	67	0
Циклический (параллельный)	241	2	8,2	111	0

В **четвёртой** главе приведено описание практического применения устройств обнаружения и исправления пакетных ошибок на основе помехоустойчивого кода (15, 8, 3) и циклического избыточного кода CRC8 для обеспечения целостности передаваемых команд подсистемы синхронизации системы управления электрофизической установки Токмак КТМ.

Разработано кодирующее устройство (рис. 5) на ПЛИС, состоящее из модуля coder, который строит помехоустойчивый код и модуля crc8, вычисляющего контрольную сумму CRC8 матричным алгоритмом.

На вход кодирующего устройства поступает пакет данных (packet [31..0]), для которого рассчитывается контрольная сумма CRC8 (всего 4 байта). Каждый байт пакета кодируется циклическим помехоустойчивым кодом (15, 8, 3), позволяющим исправить пакет ошибок длиной до 3-х бит. Таким образом, по каналу связи передается избыточный пакет (packet_sw [59..0]) длиной 60 бит, поступивший с выхода кодера. Кодирование данных (модуль ЦБС) осуществляется согласно выражению:

$$CW(0, 1) = M(0, 1) * 2^k \mid \text{mod} ((M(0, 1) * 2^k) / G(0, 1)), \text{ где:}$$

$M(0,1)$ – байт данных;

$G(0,1)$ – образующий полином 11010001;

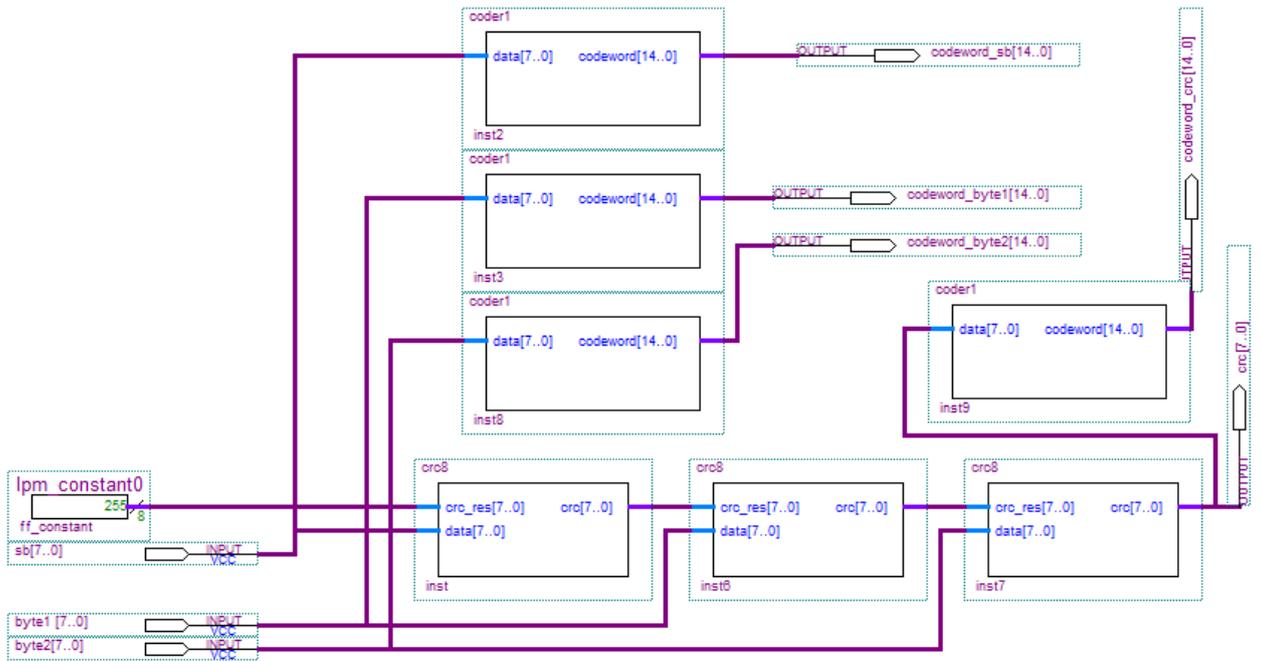


Рисунок 5 – Функциональная схема кодирующего устройства кода (15, 8, 3)

Формат закодированной команды приведен на рис. 6. Каждый байт пакета дополняется 7 контрольными битами.

8 бит	7 бит	8 бит	7 бит	8 бит	7 бит	8 бит	7 бит
START	START_K	TYPE	TYPE_K	COP	COP_K	CS	CS_K
55h	Контрольный блок	01h	Контрольный блок	Код операции	Контрольный блок	Контрольная сумма	Контрольный блок

Рисунок 6 – Пакет команды с избыточностью

Для исправления ошибок на приемной стороне разработано декодирующее устройство (рис. 7) на основе циклического помехоустойчивого кода (15, 8, 3).

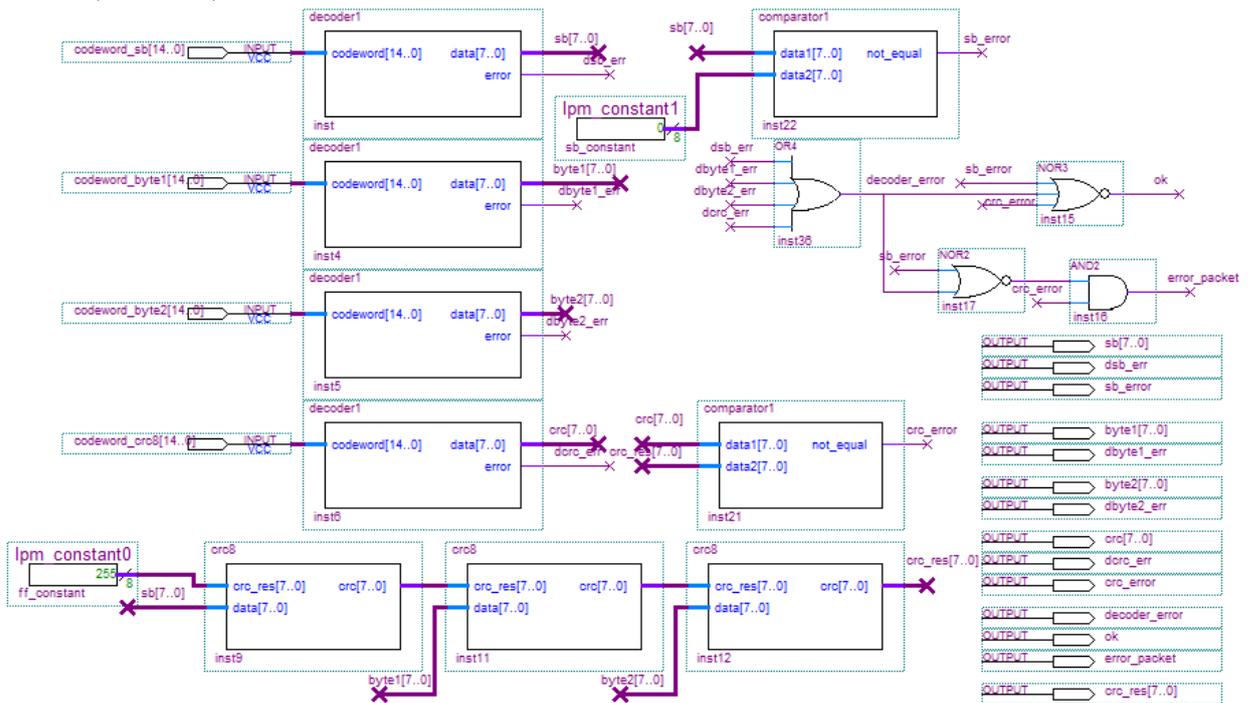


Рисунок 7 – Функциональная схема декодирующего устройства кода (15, 8, 3)

На приемной стороне декодирующее устройство получает пакет с ошибками (либо без ошибок) и осуществляет процесс исправления (декодирования). На выходе декодера образуются массив ошибок (decode_errors [3..0]) и исправленный пакет данных (packet [31..0]). Исправленный пакет разбивается на части: packet [7..0] – контрольная сумма CRC, packet [31..8] – старт-байт, packet [31..24] – данные и старт-байт. При этом packet [31..24] поступает на блок расчета CRC8.

Для проверки работоспособности кодирующего устройства на вход были поданы следующие данные: старт-байт 00h, первый байт AAh, второй байт 0Fh. На выходе кодера формируются кодовые слова длиной 15 бит. На рис. 8 представлены результаты работы кодера.

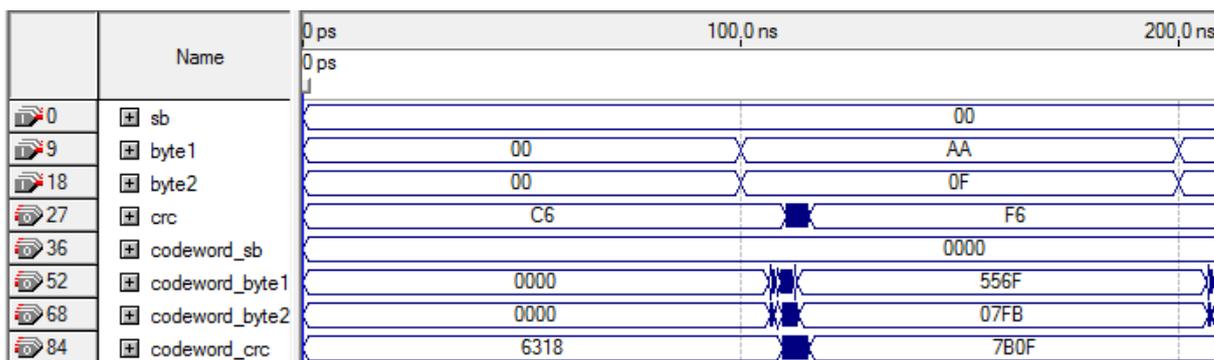


Рисунок 8 – Диаграмма работы кодирующего устройства кода (15, 8, 3)

Для проверки работы декодера сформированы кодовые слова с различными типами ошибок (см. codeword_byte1, рис. 9): кодовое слово без ошибок (556F) (101010101101111), с трёхкратной пакетной ошибкой (5568) (101010101101000), двукратной независимой (546E) (101010001101110), четырёхкратной пакетной (5560) (101010101100000). Все кодовые слова представлены в шестнадцатеричной системе счисления.

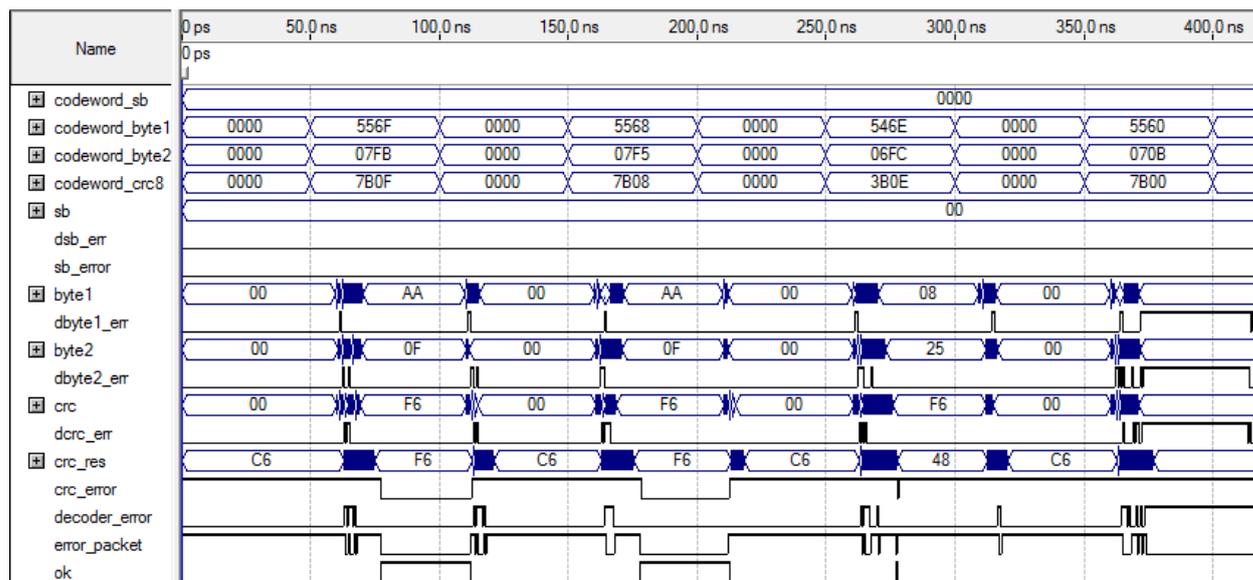


Рисунок 9 – Диаграмма работы декодирующего устройства кода (15, 8, 3)

Из результатов работы декодера следует, что для кодового слова без ошибки и с трёхкратной пакетной ошибкой на выходе устройства получены исходные байты данных, то есть ошибка исправляется. Для четырёхкратной пакетной ошибки сформирован сигнал error, сигнализирующий о неисправимой ошибке из-за превышения заданной кратности.

Для устройства кодирования на базе ПЛИС Cyclone EP1C12Q240C6 требуется 70 логических элементов ПЛИС. Время кодирования составляет 15,94 нс (асинхронный режим). Для устройства декодирования требуется 947 логических элементов ПЛИС. Время декодирования составляет 34,98 нс (асинхронный режим). При частоте синхронизации шины локального модуля синхронизации, равной 10 МГц, быстродействие декодирования составит не более 1 такта синхросигнала.

Для проверки работы разработанных устройств на ПЛИС был собран экспериментальный стенд на основе макетов SDK -6.1 с реальными каналами передачи данных. Макет SDK-6.1 включает в состав ПЛИС Cyclone EP1C3T144C8, жидкокристаллический индикатор (ЖКИ), блок движковых переключателей, блок светодиодов, последовательный порт RS-232 и порт дискретных сигналов (DIO). На рис. 10 представлен экспериментальный стенд, демонстрирующий работу декодеров на ПЛИС.

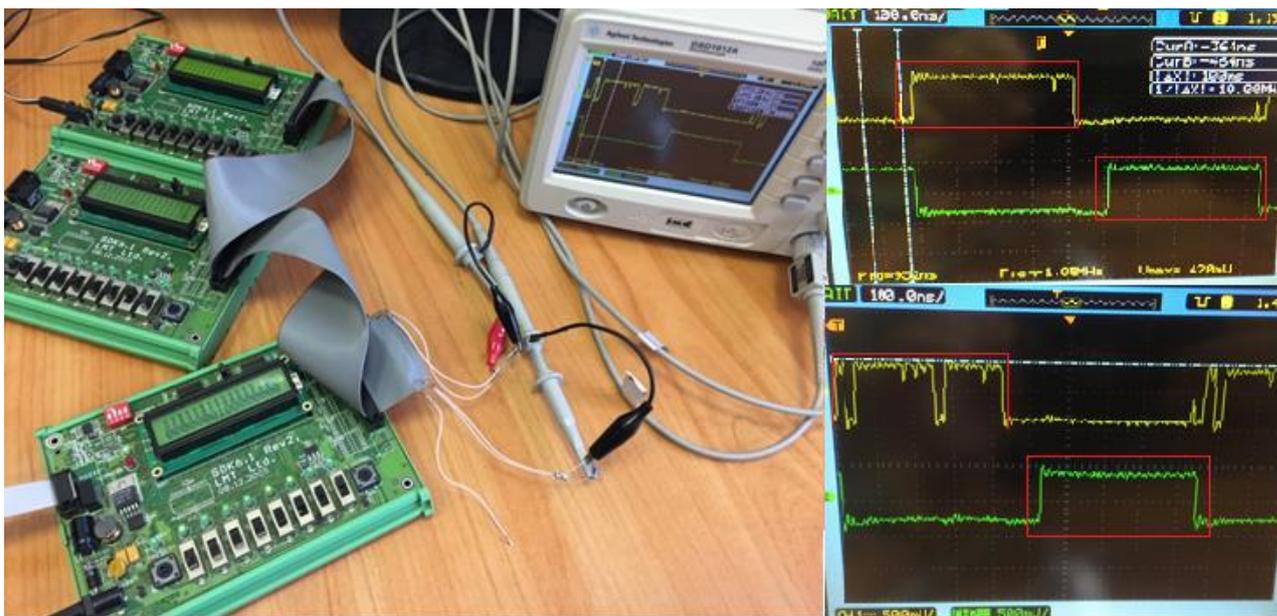


Рисунок 10 – Экспериментальный стенд для проверки работы устройств

Экспериментальный стенд состоит из 3 макетов: устройство кодирования данных, устройство, выполняющее роль канала связи с помехами и устройство декодирования данных. Все макеты соединены 40-контактным кабелем через интерфейс дискретного ввода-вывода. Для передачи данных используются LVDS-сигналы. К каналу связи подключен двухканальный осциллограф DS01012A с полосой пропускания 100 МГц. Результаты эксперимента показали возможность предложенных устройств на ПЛИС исправлять двукратные независимые и трёхкратные пакетные ошибки при передаче информации между устройствами.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

В ходе выполнения диссертационной работы получены следующие научные и практические результаты и сделаны выводы.

1. Проведён анализ существующих алгоритмов обнаружения ошибок без исправления. Обоснована актуальность разработки алгоритмов вычисления контрольной суммы CRC для систем с малым объёмом памяти и низкими вычислительными мощностями.
2. Разработаны устройства на ПЛИС вычисления циклических избыточных кодов CRC8 и CRC32 на основе матричного алгоритма с модификациями, позволяющими увеличить размер блока данных, обрабатываемого за итерацию. Установлено, что матричный алгоритм при программной реализации требует до 32 раз меньший объем памяти для хранения предвычисленных значений (в зависимости от размера блока, обрабатываемого за итерацию), чем при реализации табличного алгоритма. Результаты исследования аппаратных реализаций показали, что при использовании матричного алгоритма быстродействие устройств увеличивается в 2 раза, а изменение длины блока, обрабатываемого за итерацию, позволяет увеличить быстродействие до 8–10 раз по сравнению с устройствами на основе табличного алгоритма.
3. Проведён анализ существующих кодов БЧХ, применяемых для исправления независимых ошибок. Обоснована актуальность разработки алгоритма поиска образующих полиномов для построения циклических помехоустойчивых кодов более эффективных, чем коды БЧХ.
4. Разработан и реализован алгоритм поиска образующих полиномов, адаптированный для параллельных вычислений и отличающийся от известных тем, что позволяет находить полиномы для построения циклических помехоустойчивых кодов более эффективных по скорости кода, чем коды БЧХ.
5. Поставлен компьютерный эксперимент по поиску образующих полиномов с применением технологий параллельных вычислений для построения кодов, исправляющих независимые и пакетные ошибки. Установлено, что разработанный алгоритм поиска позволяет находить полиномы для построения кодов длиной до 5 бит короче, чем коды БЧХ, при длине информационного блока до 32 бит и кратности исправляемых ошибок до 4. Установлено, что применение технологий параллельных вычислений целесообразно для большинства из исследуемых длин информационного блока.
6. Проведено исследование аппаратных реализаций на ПЛИС модифицированного циклического алгоритма декодирования для предложенного кода (17, 9, 5), являющегося более эффективным кодом по скорости, чем код БЧХ (15, 7, 5) и позволяющего реализовать более эффективные по быстродействию и аппаратным затратам устройства исправления ошибок, чем с применением укороченного кода БЧХ (19, 9, 5).

7. Разработаны декодирующие устройства циклического помехоустойчивого кода (15, 8, 3) на ПЛИС, исправляющие пакетные ошибки длиной до 3 бит с применением табличного и циклического алгоритмов декодирования. Установлено, что декодеры обладают лучшим быстродействием и меньшими аппаратными затратами, чем декодеры БЧХ кода (15, 7, 5) с аналогичной длиной кодового слова.
8. Разработано устройство исправления пакетных ошибок длиной до 3 бит с применением модифицированного циклического алгоритма декодирования для решения задачи по обеспечению надежности передаваемых команд подсистемы синхронизации системы управления электрофизической установки Токамак КТМ.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

В рецензируемых изданиях, рекомендованных ВАК РФ

1. Мыцко, Е. А. Исследование аппаратных реализаций табличного и матричного алгоритмов вычисления CRC32 / Е. А. Мыцко, А. Н. Мальчуков // Известия Томского политехнического университета. – 2013. – Т. 322, № 5. – С. 182–186.
2. Мыцко, Е. А. Исследование программных реализаций алгоритмов вычисления CRC совместимых с PKZIP, WINRAR, ETHERNET / Е. А. Мыцко, А. Н. Мальчуков // Известия Томского политехнического университета. – 2013. – Т. 322, № 5. – С. 170–175.
3. Мыцко Е. А., Программная реализация алгоритма поиска образующих полиномов с применением технологий OpenMP и MPI / Е.А. Мыцко, А. Н. Мальчуков // Современные проблемы науки и образования. – 2014. – № 6. – С. 1–8.
4. Мыцко Е. А. Разработка структуры быстродействующего декодера БЧХ-кода (15,7,5) на основе метода циклического декодирования / А. Н. Мальчуков, С. Е. Рыжова, И. В. Зоев, В. Л. Ким // Прикладная информатика. – 2017. – Т. 12, № 2(68). – С. 72–78.
5. Мыцко Е. А. Исследование алгоритмов вычисления контрольной суммы CRC8 в микропроцессорных системах при дефиците ресурсов / Е.А. Мыцко, А. Н. Мальчуков, С. Д. Иванов // Приборы и системы. Управление, контроль, диагностика. – 2018 – № 6. – С. 22–29.
6. Мыцко Е. А. Исследование программных реализаций декодеров циклических помехоустойчивых кодов, исправляющих пакетные ошибки при дефиците ресурсов / Е.А. Мыцко, А. Н. Мальчуков, С. Д. Иванов // Приборы и системы. Управление, контроль, диагностика. – 2018. – № 9. – С. 27–36.

В изданиях, индексируемых в SCOPUS и Web of Science

7. Mytsko E. A. Application of parallel computing technology OpenMP to search for the generator polynomials / Е.А. Mytsko, А.Н Malchukov // Mechanical Engineering, Automation and Control Systems: Proceedings of International

- Conference, Tomsk, October 16-18, 2014. – Tomsk: TPU Publishing House, 2014. – P. 1–5.
8. Mytsko E.A. Adaptation of technology MPI and OpenMP to search for the generators polynomials / E.A. Mytsko, A.N. Malchukov // Proceedings of 9th International Forum on Strategic Technology (IFOST-2014), Chittagong, Bangladesh: IEEE Press, 2014. – P. 5–8.
 9. Mytsko E. A. Structure Development of the BCH Code High-speed Decoder Based on the Cyclic Decoding Method / E.A. Mytsko, A.N. Malchukov, S.E. Ryzhova, V.L. Kim // 2015 International Conference on Applied Mechanics and Mechatronics Engineering (AMME 2015), Bangkok, October 25-26, 2015. – Lancaster: DEStech Publications Inc., 2015. – P. 175–178.
 10. Mytsko E.A. Fast Decoder of BCH Code with Cyclic Decoding Method / E.A. Mytsko, A.N. Malchukov, I.V. Novozhilov, V.L. Kim // Proceedings of International Siberian Conference on Control and Communications (SIBCON - 2016): Москва: ВШЭ, 2016. – P. 1–4.
 11. Mytsko E. A. A study of hardware implementations of the CRC computation algorithms / E.A. Mytsko, A.N. Malchukov, S.E. Ryzhova, V.L. Kim // MATEC Web of Conferences. – 2016. – Vol. 48, Article number 04001. – P. 1–7.
 12. Mytsko E. A. Software Implementation Research of CRC Computation Algorithms Compatible with PKZIP, WINRAR, ETHERNET / E.A. Mytsko, A.N. Malchukov, V.L. Kim, A.N. Osokin, I.V. Zoev, S.E. Ryzhova // Advances in Computer Science Research. – 2016. – Vol. 51. – P 134–138.
 13. Mytsko E. A. FPGA design of the fast decoder for burst errors correction / E.A. Mytsko, A.N. Malchukov, I.V. Zoev, S.E. Ryzhova, V.L. Kim // Journal of Physics: Conference Series. – 2017. – Vol. 803, Article Number 012105. – P. 1–6.

Доклады на конференциях

14. Мыцко Е. А. Реализация устройства контроля правильности передачи данных в системе синхронизации и противоаварийной защиты установки токамак КТМ / Е.А. Мыцко, А. Н. Мальчуков // Молодежь и современные информационные технологии: сборник трудов XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых, 9-13 ноября 2015 г., г. Томск, в 2 т. 2015. – Т. 1. – С. 320–321.
15. Мыцко, Е. А. Применение технологий параллельных вычислений для поиска образующих полиномов, используемых при построении эффективных помехоустойчивых кодов / Е.А. Мыцко // Научная сессия ТУСУР-2017: материалы Международной научно-технической конференции студентов, аспирантов и молодых ученых, посвященной 55-летию ТУСУРа, 10-12 мая 2017 г., г. Томск в 8 ч. – 2017. – Ч. 5. – С. 141–143.
16. Мыцко, Е. А. Применение технологий параллельных вычислений в задаче поиска образующих полиномов для построения эффективных помехоустойчивых кодов / Е.А. Мыцко // Материалы 55-й Международной

- научной студенческой конференции МНСК-2017, 16-20 апреля 2017 г. Информационные технологии. – Новосибирск: Изд-во НГУ, 2017. – С. 135.
17. Мыцко, Е. А. Исследование программных и аппаратных реализаций алгоритмов вычисления контрольной суммы CRC / Е.А. Мыцко // Наука будущего - наука молодых: сборник тезисов докладов на конференции III Всероссийский научный форум, Нижний Новгород, 12-14 сентября 2017 г. – 2017. – Т. 1. – С. 127–128.
 18. Мыцко, Е. А. Исследование алгоритмов вычисления контрольной суммы CRC8 для микроконтроллерной системы измерения температуры / Е.А. Мыцко // Наука. Технологии. Инновации: сборник научных трудов: в 10 т., Новосибирск, 4-8 Декабря 2017. – Новосибирск: НГТУ, 2017. – Т. 1 – С. 90–94.

Прочие публикации

19. Мыцко Е. А. Поиск образующих полиномов для построения декодеров, исправляющих пакетные ошибки с применением технологий параллельных вычислений / Е.А. Мыцко // Высокопроизводительные вычислительные системы и технологии. – 2018. – № 1 (8). – С. 97–101.
20. Мыцко Е. А. Сравнение аппаратных реализаций комбинированного метода декодирования на примере кода (17, 9) / С.Е. Рыжова, Е. А. Мыцко // Высокопроизводительные вычислительные системы и технологии, 2018. – № 1(8). – С. 24–28.
21. Мыцко Е. А. Проектирование и реализация устройств на ПЛИС для исправления независимых ошибок с применением циклических помехоустойчивых кодов (17, 9, 5) и (19, 9, 5) / Е.А. Мыцко // Высокопроизводительные вычислительные системы и технологии. – 2018. – № 2 (9). – С. 29–34.

Свидетельства о регистрации программ для ЭВМ

22. Мыцко Е. А., Мальчуков А. Н., Рыжова С. Е., Зоев И. Вычисление контрольной суммы CRC32 матричным алгоритмом // Свидетельство о государственной регистрации программы для ЭВМ № 2016617787.
23. Рыжова С. Е., Мальчуков А. Н., Мыцко Е. А., Зоев И. В. Быстродействующее декодирование кода Боуза-Чоудхури-Хоквингема (15, 7, 5) // Свидетельство о государственной регистрации программы для ЭВМ № 2016662525.
24. Рыжова С. Е., Мыцко Е. А., Мальчуков А. Н. Быстродействующий декодер кода Боуза-Чоудхури-Хоквингема (15,5,7), исправляющий до 3-х независимых ошибок // Свидетельство о государственной регистрации программы для ЭВМ № 2018663370.
25. Мыцко Е. А. Поиск образующих полиномов для построения циклических помехоустойчивых кодов с применением технологий параллельных вычислений // Свидетельство о государственной регистрации программы для ЭВМ № 2018665738.