Министерство образования Республики Беларусь

Учреждение образования

«Белорусский Государственный технологический университет»

Кафедра программной инженерии

Отчет по лабораторной работе № 3

**«Исследование циклического избыточного кода (CRC)»**

Дисциплина «Компьютерные сети и симтемы»

Специальность «Програмная инженерия»

**Выполнил**:

Студент 1 курса 10 группы 1 подгруппы ФИТ

Макаревич Кирилл Витальевич

2025

**Отчет о лабораторной работе №3**

**Цель работы:** получение навыков использования циклического избыточного кода (CRC).

**1. Краткие теоретические сведения**

При передаче сигнала через любой канал связи возможно возникновение ошибок, которые могут приводить к искажению переносимой информации. Существует много методов для исправления подобных ошибок, но прежде чем исправлять, необходимо эти ошибки обнаружить. Для этого также существуют определенные методы, основанные на избыточности передаваемой информации, что позволяет не только выявлять наличие факта искажения информации, но и в ряде случаев устранять эти искажения.

Наиболее известные из методов обнаружения ошибок передачи данных являются:

* Посимвольный контроль четности, называемый также поперечным
* Поблочный контроль четности, называемый продольным
* Вычисление контрольных сумм
* Контроль циклически избыточным кодом — CRC (Cyclical Redundancy Check).

Циклический избыточный код (англ. Cyclic redundancy code, CRC) — алгоритм вычисления контрольной суммы, предназначенный для проверки целостности передаваемых данных. Алгоритм CRC обнаруживает все одиночные ошибки, двойные ошибки и ошибки в нечетном числе битов. Понятие циклических кодов достаточно широкое, однако на практике его обычно используют для обозначения только одной разновидности, использующей циклический контроль (проверку) избыточности. В связи с этим в англоязычной литературе CRC часто расшифровывается как Cyclic Redundancy Check.

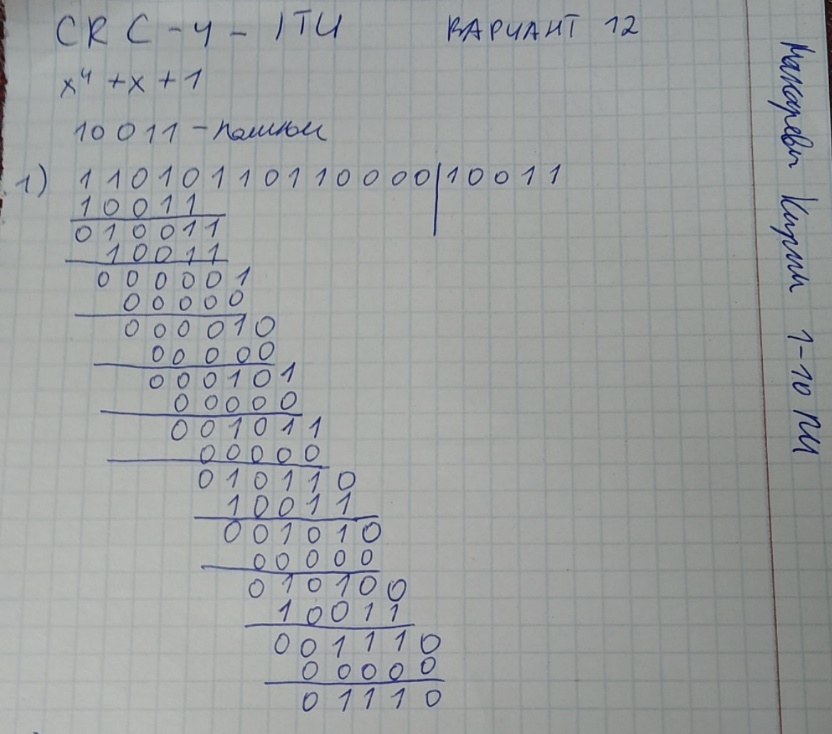
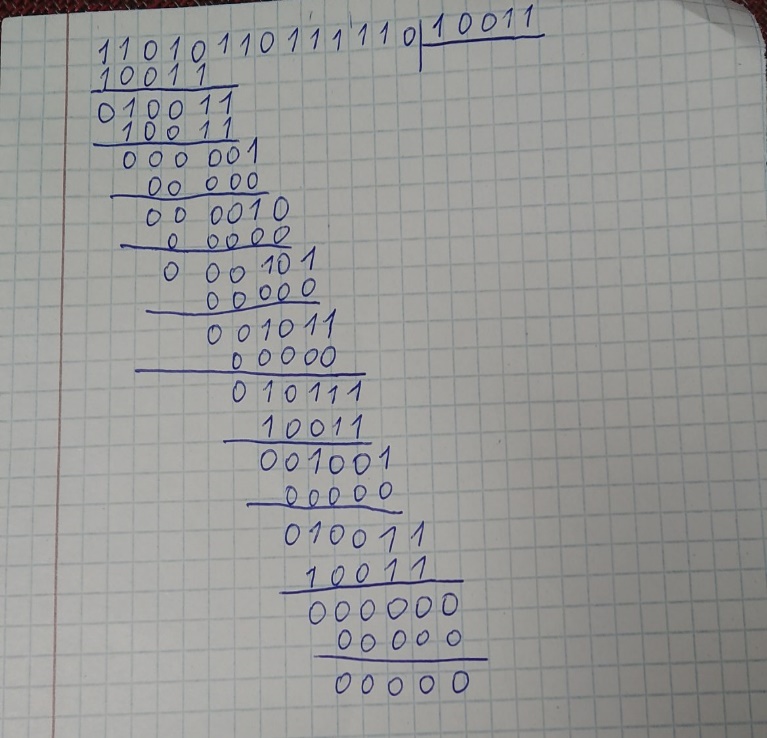
CRC некоторой последовательности вычисляется на основании другой (исходной) битовой последовательности. Главная особенность (и практическая значимость) значения CRC состоит в том, что оно однозначно идентифицирует исходную битовую последовательность и поэтому используется в различных протоколах связи, а также для проверки целостности блоков данных, передаваемых различными устройствами. Благодаря относительной простоте алгоритм вычисления CRC часто реализуется на аппаратном уровне.

Основная идея вычисления CRC заключается в следующем. Исходная последовательность битов, которой могут быть и огромный файл, и текст размером несколько слов и даже символов, представляется единой последовательностью битов. Эта последовательность делится на некоторое фиксированное двоичное число (полином, CRC-полином, генераторный полином, англ. generator polinomial). Интерес представляет остаток от этого деления, который и является значением CRC. Все, что теперь требуется, — это некоторым образом запомнить его и передать вместе с исходной последовательностью. Приемник данной информации всегда может таким же образом выполнить деление и сравнить его остаток с исходным значением CRC. Если они равны, то считается, что исходное сообщение не повреждено, и т. д.

Простейший алгоритм вычисления CRC включает следующие шаги:

1. Исходная битовая последовательность дополняется нулями справа (количество нулей равно степени генераторного полинома).
2. Дополненная последовательность делится на генераторный полином с использованием CRC-арифметики (операция XOR).
3. Остаток от деления — это CRC, который добавляется к исходной последовательности.
4. На стороне приемника CRC проверяется путем повторного деления всей последовательности (включая CRC) на тот же полином. Если остаток равен нулю, данные считаются корректными.

**2. Практическая часть**

1. **Контрольные вопросы**

**1. Как расшифровывается и переводится CRC?**

CRC расшифровывается как \*\*Cyclic Redundancy Check\*\*, что переводится как «циклический избыточный код» или «циклическая проверка избыточности». Это метод обнаружения ошибок в данных.

**2. Для каких целей обычно используется CRC?**

CRC применяется для:

- Обнаружения случайных ошибок при передаче или хранении данных (например, в сетевых протоколах, файловых системах, архивах).

- Проверки целостности данных (например, в контрольных суммах файлов, при записи на диски).

- Выявления искажений, вызванных помехами, сбоями оборудования или программными ошибками.

**3. Определение CRC.**

CRC — это алгоритмический метод, генерирующий короткое фиксированное значение (контрольную сумму) на основе данных. Он использует циклические коды, основанные на делении полиномов, для детектирования случайных изменений в исходной информации.

**4. Принципиальная особенность CRC-арифметики.**

Основана на полиномиальной арифметике в двоичном поле. Операции выполняются над битовыми последовательностями, интерпретируемыми как коэффициенты полиномов. Вместо арифметического деления используется побитовый XOR, что делает вычисления быстрыми и эффективными для аппаратной реализации.

**5. Простейший алгоритм вычисления CRC.**

1. Выбор полинома: Фиксируется генераторный полином (например, `CRC-16: x¹⁶ + x¹⁵ + x² + 1`, записывается как битовая маска `0x8005`).

2. Инициализация: Данные дополняются нулями справа (количество нулей равно степени полинома).

3. Деление:

- Данные обрабатываются бит за битом (сдвиг влево).

- Если старший бит равен 1, выполняется XOR с полиномом.

- Процесс повторяется до конца данных.

4. Результат: Остаток от деления (после обработки всех битов) — это CRC, который добавляется к исходным данным.

Пример:

Для данных `1101` и полинома `1011`:

1. Данные дополняются до `1101000`.

2. Последовательное деление с XOR дает остаток `001`, который становится CRC.

3. Итоговое сообщение: `1101000 001`.

1. **Выводы**

Циклический избыточный код (CRC) является мощным и широко используемым методом обнаружения ошибок при передаче данных. Он основан на избыточности информации, что позволяет не только выявлять факт искажения данных, но и в ряде случаев устранять эти искажения. Основное преимущество CRC заключается в его высокой эффективности при обнаружении одиночных, двойных и ошибок в нечетном числе битов, что делает его незаменимым в современных системах связи и передачи данных.

CRC использует полиномиальную арифметику по модулю 2, где сложение и вычитание заменяются операцией XOR. Это упрощает вычисления и позволяет эффективно реализовать алгоритм как на программном, так и на аппаратном уровне. Процесс вычисления CRC включает деление битовой последовательности на генераторный полином, а остаток от этого деления (CRC) добавляется к исходным данным для последующей проверки на стороне приемника.

Таким образом, CRC является важным инструментом для обеспечения надежности передачи данных, широко применяемым в различных протоколах связи, таких как Ethernet, USB, Bluetooth и многих других. Его простота и высокая эффективность делают его одним из наиболее популярных методов контроля целостности данных.