Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



**Отчет**

**По домашнему заданию №2**

**По курсу «Сети и телекоммуникации»**

**Вариант 9**

**ИСПОЛНИТЕЛЬ:**

Группа ИУ5-53Б

Кузнецов В.А.

"6"декабря 2023 г.

**ПРЕПОДАВАТЕЛЬ:**

\_\_Галкин В.А.\_\_

"    "            2023 г.

Москва 2023

# Метод решения задачи для варианта задания

|  |  |  |  |
| --- | --- | --- | --- |
| № варианта | Информационный вектор | Код |  |
| 9 | 0010 | Ц [7,4] | 𝐶0 |

Для решения задачи оценки корректирующей способности кода нужно перебрать все возможные значения ошибок , сгруппировав их по классу (по числу единиц). Для каждой группы посчитать сколько ошибок удалось обнаружить. Обнаружение ошибки происходит при помощи вычисления синдрома, произведя над вектором ошибки операцию деления на образующий полином g(x) = x3 + x + 1 (10112).

Всего есть 7 классов ошибок: от ошибок разрядности 1 до ошибок разрядности 7. Число ошибок в каждом разряде определяется по формуле: , где i – кратность ошибки. Число ошибок в каждом классе подсчитано на компьютере и приведено в результирующей таблице.

Имитируя канал связи, передавать по нему информационный вектор в закодированном виде (10110), накладывая поочередно все вектора ошибок. Имитируя приемник, расшифровать сообщение, определив ошибку. Если вектор синдрома-ошибки отличен от 0, то ошибка обнаружена.

После перебора ошибок данного класса рассчитать обнаруживающую способность и записать ее в результирующую таблицу, выразив в процентах.

Для решения поставленной задачи была написана программа на языке JavaScript с использованием побитовых операция и визуализацией данных с помощью HTML-страницы.

# Реализация алгоритмов кодирования, декодирования, модели канала связи и вычисления обнаруживающей способности кода для ошибок всех возможных кратностей.

Передатчик имитируется частью программного кода (переменной), содержащей исходное закодированное сообщение 0010.

Трансфер по каналу связи имитируется частью программного кода (циклом), в котором на исходное закодированное сообщение накладываются по очереди вектора ошибок каждого класса.

Приемник имитируется частью программного кода (функцией), которая делит входящий циклический код на образующий полином, и по полученному синдрому определяет наличие ошибки.

**Реализация алгоритма кодирования**

Функция **code** реализует алгоритм кодирования циклическим [n,k]-кодом. Она принимает на вход информационный вектор **m**, порождающий вектор **g, n, k.** Результатом функции является циклический [n,k]-код.

function code(m, g, n, k) {

    let tmp = m.slice().concat(Array(n - k).fill(0));

    let reminder = divide(tmp, g)

    return m.slice().concat(reminder)

}

function divide(a, b) {

    for (let i = 0; i <= (a.length - b.length); i++) {

        if (a[i] === 0) {

            continue

        }

        for (let j = 0; j < b.length; j++) {

            a[i + j] = a[i + j] ^ b[j]

        }

    }

    return a.slice(-(b.length - 1))

}

**Реализация алгоритма декодирования:**

Функция **checkError** реализует алгоритм декодирования циклического [n, k ]-

кода. Она принимает на вход циклический [n, k]-код. Результатом функции

является наличие ошибки в коде.

function checkError(r, g) {

    let s = divide(r, g)

    for (let i of s) {

        if (i !== 0) {

            return true

        }

    }

    return false

}

**Реализация модели канала связи**

Функция **error** реализует модель канала связи. Она принимает на вход 2 параметра: кодовый вектор и вектор ошибки. Результатом функции является кодовый вектор, принятый выходе канала связи.

function code(m, g, n, k) {

    let tmp = m.slice().concat(Array(n - k).fill(0));

    let reminder = divide(tmp, g)

    return m.slice().concat(reminder)

}

**Реализация вычисления обнаруживающей способности кода для ошибок всех возможных кратностей:**

Вычисление обнаруживающей способности кода для ошибок всех

возможных кратностей выполняется в рекурсивной функции **generateErrors**. Для каждого вектора ошибки текущей кратности с помощью функций **error** и **checkError** вычисляются соответственно принятый на выходе канала связи кодовый вектор и наличие ошибки по синдрому. Результаты вычислений записываются в таблицу данных.

function generateErrors(n, k, callback) {

    function backtrack(current, onesCount, remainingZeros) {

        if (onesCount + remainingZeros === 0) {

            callback(current);

            return;

        }

        if (onesCount > 0) {

            backtrack(current.slice().concat([1]), onesCount - 1, remainingZeros);

        }

        if (remainingZeros > 0) {

            backtrack(current.slice().concat([0]), onesCount, remainingZeros - 1);

        }

    }

    backtrack([], k, n - k);

    return;

}

for (let i = 1; i <= n; i++) {

        let errors = 0

        let detected = 0

        generateErrors(n, i, (e) => {

            errors += 1

            let r = error(coded, e);

            if (checkError(coded, e, g)) {

                detected += 1

            }

        })

        let newRow = table.insertRow();

        let cell1 = newRow.insertCell(0);

        cell1.textContent = i.toString()

        let cell2 = newRow.insertCell(1);

        cell2.textContent = errors.toString()

        let cell3 = newRow.insertCell(2);

        cell3.textContent = detected.toString()

        let cell4 = newRow.insertCell(3);

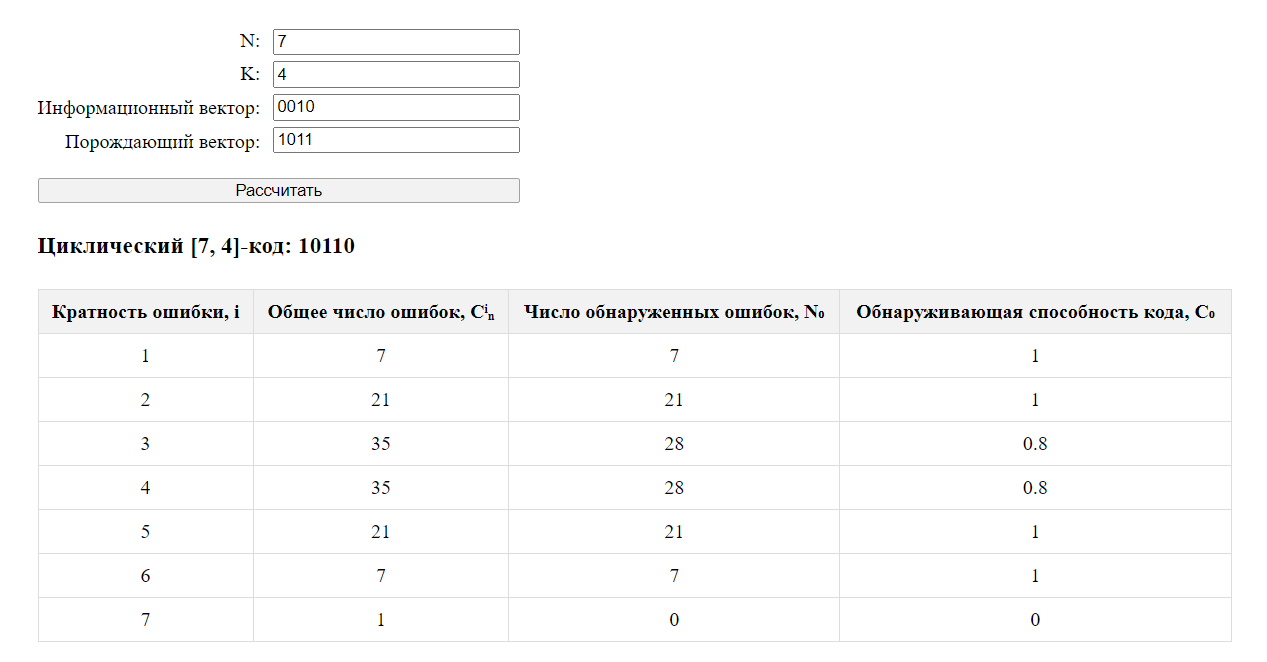
        cell4.textContent = (detected / errors).toString()

    }

1. **Полный текст** **программы**

<https://github.com/Qesait/st-dz2>

1. **Заполненная программно таблица результатов**

****

# Выводы

По таблице результатов видно, что только при кратности ошибки равной 7 обнаруживающая способность циклического [7,4]-кода равна 0 и он не способен обнаружить ошибку. При кратностях ошибки равными 1, 2, 5 и 6 обнаруживающая способность кода равна 1 и ошибку всегда можно обнаружить. При остальных кратностях ошибки обнаруживающая способность кода равна 0,8, что даёт высокие шансы обнаружить ошибку. Таким образом, циклический подходит для обнаружения ошибок кратностью до 6.

# Список используемой литературы

* Телекоммуникации и сети. / В.А.Галкин, Ю.А.Григорьев Учебное пособие для вузов.- М.:Из-во МГТУ им.Н.Э.Баумана 2003 г.
* Методическое пособие по выполнению домашнего задания по дисциплине «Сети и телекоммуникации» / Галкин В.А. М.: МГТУ им. Н.Э.Баумана. 2018 г.
* Научная библиотека scask. К.К. Васильев, В.А. Глушков, А.В. Дормидонтов, А.Г. Нестеренко. «Теория электрической связи: учебное пособие». Ульяновск: УлГТУ, 2008. – 452 с. [214] <https://scask.ru/a_book_tec.php?id=82>