-МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

Лабораторная работа № 8

по теме: «Перемежитель**»**

Выполнил: студент ФИТ

3 курс, 1группа

Калинов И.О

Проверил:

доцент каф. ИСиТ

Шиман Д.В.

Минск, 2016

Оглавление

[Постановка задачи 3](#_Toc470227605)

[Общие теоретические сведения 4](#_Toc470227606)

[Реализация 5](#_Toc470227607)

[Результат программы 7](#_Toc470227608)

[Вывод 10](#_Toc470227609)

# **Постановка задачи**

Изучить механизм работы блоковых перемежителей; разработать модель кодирования, перемежения, передачи, деперемежения и декодирования сообщений с использованием произвольного метода перемежения и кодирования; изучить действия алгоритма при передаче сообщения без ошибок и с групповыми ошибками разной длины.

# **Общие теоретические сведения**

Большинство хорошо известных кодов, которые были разработаны для увеличения надёжности при передачи информации, являются эффективными, когда ошибки, вызванные каналом, статистически независимы. Однако, имеются каналы, в которых появляются пакеты ошибок Замирания сигнала, обусловленные меняющимся во времени много путевым распространением волн, часто вызывают снижение уровня сигнала ниже уровня шума, что приводит к большому числу ошибок. Второй пример – это класс каналов магнитной записи (ленточной или дисковой), в которых дефекты в записывающей среде приводят к пачкам ошибок. Такие группы ошибок обычно не исправляются кодами, оптимально рассчитанными для статистически независимых ошибок.

При передаче информационного сообщения Xk по каналу в сообщении могут возникать групповые ошибки. Они выражаются в инверсии нескольких последовательно передающихся битов сообщения. Для того чтобы обнаруживать исправлять возникающие групповые ошибки, используется избыточное кодирование и перемежение. Избыточное кодирование выполняется над блоками произвольной длины, на которые разбивается информационное сообщение. Полученные блоки с проверочными символами длиной n объединяются в кодовое слово Xn. Перемежение применяется после кодирования и заключается в изменении последовательности следования битов кодового слова Xn. В зависимости от наиболее часто возникающих ошибок разрабатываются соответствующие методы перемежения. Наиболее эффективно располагать подряд биты всех блоков. Повышенная помехоустойчивость может быть обеспечена разнесением проверочных битов блоков в разные части перемеженного сообщения. В некоторых случаях это позволит получить несколько одинарных ошибок в проверочных битах одного блока без повреждения информации.

# **Реализация**

Функция кодирования

textBox4.Text = "";

int k = R;

int r = Convert.ToInt32(Math.Round(Math.Log(k) + 2));//определения количества проверочных символов

R = Convert.ToInt32(textBox3.Text);

int[] Soobshch = new int[textBox2.TextLength];//чтение данных

int[,] Matrica = new int[k + r, K/R];//делим на определенные длины

for(int i=0;i<textBox2.TextLength;i++)//запись поэлементно

{

if(textBox2.Text[i]==48)

{

Soobshch[i] = 0;

}

if(textBox2.Text[i]==49)

{

Soobshch[i] = 1;

}

}

int[] inf\_soob = new int[k];

int[,] ProvMatHeming = Metods.HemingKodprMat(r, k, inf\_soob);//получение проверочной матрицы с помощью кода хэмминга

richTextBox1.Clear();//чистка

richTextBox1.Text += "Проверочная матрица:\n";//вывод проверочной матрицы

for (int i = 0; i < r; i++)

{

for (int j = 0; j < k + r; j++)

{

richTextBox1.Text += Convert.ToString(ProvMatHeming[j, i]) + " ";//добавление пробелов послеэлементов

}

richTextBox1.Text += Environment.NewLine;

}

Matrica = Metods.StroimMatr(Soobshch,K, R);//получение общей матрицы

Функция декодирования

Первоначально выполняются действия схожи с принципом кодирования.

}

richTextBox2.Clear();//чистка

richTextBox2.Text += "Матрица по принятову кодовому слову:\n";

for(int i=0;i<K/R;i++)

{

for(int j=0;j<k+r;j++)

{

richTextBox2.Text += Convert.ToString(PrishMatr[j, i]) + " ";//вывод матрицы

}

richTextBox2.Text += "\n";

}

//создание переменных для работы с проверкой и исправления ошибок

int[] sindrom = new int[r];

int[] soobshcenka = new int[k + r];

int[] inf\_soob = new int[k];

int[,] ProvMatHeming = Metods.HemingKodprMat(r, k, inf\_soob);//получение проверочной матрицы по коду хэмминга

newsindrom[i, j] = sindrom[j];//нужно для хранения синдромов

richTextBox2.Text += Convert.ToString(sindrom[j]);

}

richTextBox2.Text += Environment.NewLine;//вывод элементов

}

int[,] ProvMatHeming1 = Metods.HemingKodprMat(r, k, inf\_soob);//получение проверочной матрицы

IsprMatrica = Metods.IspravlMatr(peremejSoobshch, R, K);//функции возвращает исправленную матрицу

# Результат программы

Демонстрация работы программы при отсутствии ошибки (рис.1)

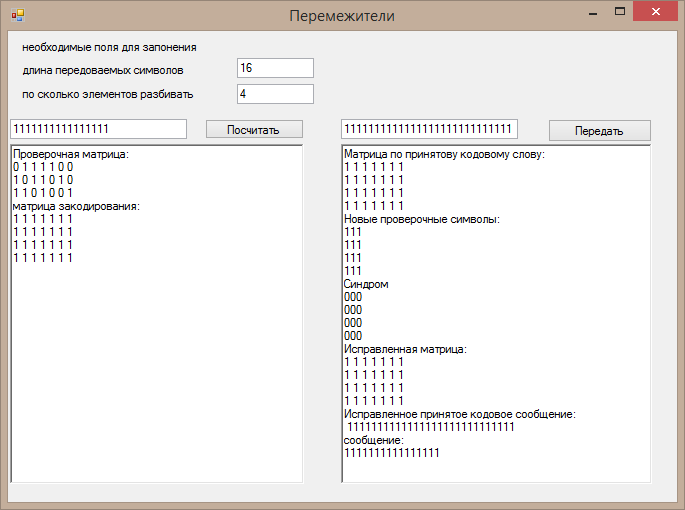


Рисунок 1 – без ошибок

Демонстрация работы программы при одной ошибки (рис.2)

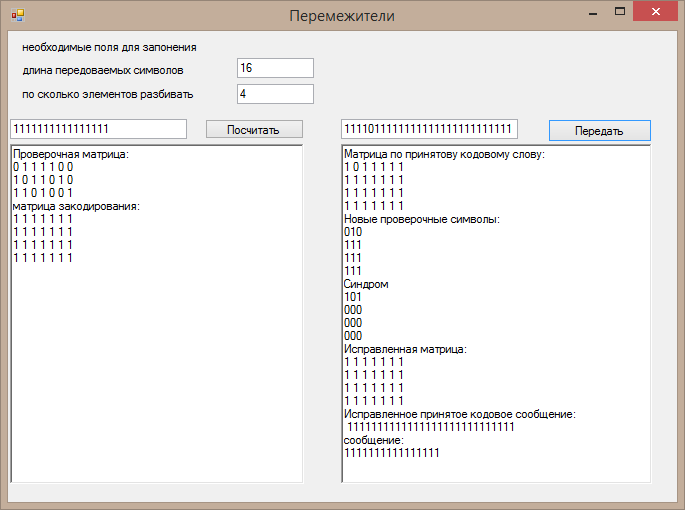


Рисунок 2 – с одной ошибкой

Демонстрация работы программы при двойной ошибки (рис.3)

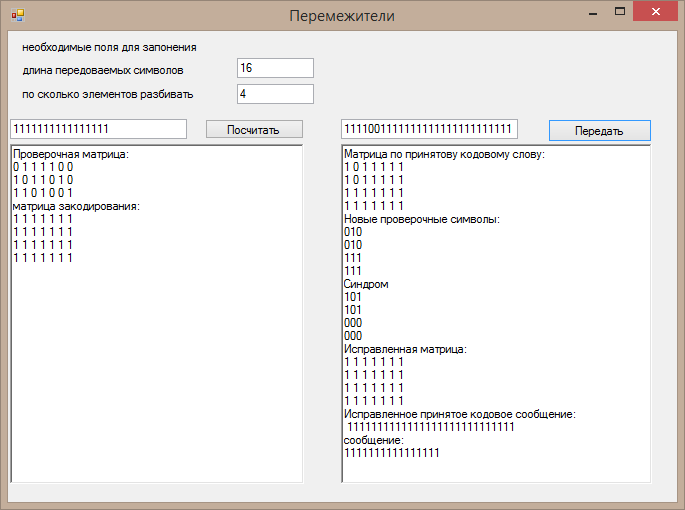


Рисунок 2 – с двумя ашибкой

# Вывод

В ходе выполнения лабораторной работы было разработано программное средство, которое реализовывает перемежение и кодирование сообщения. Позволяет исправлять количество групповых ошибок равное числу строк.