|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Мытищинский филиал**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Космический

КАФЕДРА «Прикладная математика, информатика и вычислительная техника» К3-МФ

**Лабораторная работа №4**

*ПО ДИСЦИПЛИНЕ:*

***Сети ЭВМ и телекоммуникации***

***НА ТЕМУ:***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Логическое кодирование: \_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_скремблирование\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_К3-53Б\_\_  **\_\_\_\_\_\_\_\_\_\_** Ярославцев Егор Викторович

(Группа) (Подпись, дата) (И.О.Фамилия)

* Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_** Подворная Анастасия Витальевна

(Подпись, дата) (И.О.Фамилия)

*2021 г.*

***Задача***

Написать программу скремблирования и дескремблирования и прогнать через неё сообщение из 2 лабораторной работы. Для полученного нового сообщения выполнить физическое кодирование с использованием способов кодирования, использованных на втором этапе. Результаты кодирования изобразить в виде временных диаграмм.

***Исходные данные***

Из таблицы **ЦЮА** будет представлено в шестнадцатеричном коде: **D6 00 DE FF C0**

В двоичном коде**: 11010110 00000000 11011110 11111111 11000000**

Длина сообщения: **5 байт (40 бит)**

**Алгоритм скремблирования/дескремблирования**



**Листинг программы**

#include <iostream>  
using namespace std;  
  
string Scrambling(const string& code)  
{  
 string res;  
  
 for(auto i = 0; i < code.length(); i++)  
 {  
 bool temp;  
 if(i < 5)  
 temp = code[i] - '0';  
  
 else if(i < 9)  
 temp = code[i] - '0' ^ res[i-5] - '0';  
  
 else if(i < 11)  
 temp = code[i] - '0' ^ res[i-5] - '0' ^ res[i-9] - '0' ;  
  
 else  
 temp = code[i] - '0' ^ res[i-5] - '0' ^ res[i-9] - '0' ^ res[i-11] - '0';  
  
 res += std::to\_string(temp);  
 }  
 return res;  
}  
  
string deScrambling(const string& code)  
{  
 string res;  
  
 for(auto i = 0; i < code.length(); i++)  
 {  
 bool temp;  
 if(i < 5)  
 temp = code[i] - '0';  
  
 else if(i < 9)  
 temp = code[i] - '0' ^ code[i-5] - '0';  
  
 else if(i < 11)  
 temp = code[i] - '0' ^ code[i-5] - '0' ^ code[i-9] - '0' ;  
  
 else  
 temp = code[i] - '0' ^ code[i-5] - '0' ^ code[i-9] - '0' ^ code[i-11] - '0';  
  
 res += to\_string(temp);  
 }  
 return res;  
}  
  
void check(const string& A)  
{  
 if(A == deScrambling(Scrambling(A)))  
 cout << "OK";  
 else  
 cout << "fignia(";  
}  
  
int main()  
{  
 string test = "1101011000000000110111101111111111000000";  
 cout << "Code:\t\t" << test << "\n";  
 cout << "Scrambling:\t" << Scrambling(test) << "\n";  
 printf("16bit res:\t%llX\n", strtoull(Scrambling(test).c\_str(), **NULL**, 2) );  
 cout << "deScrambling:\t" << deScrambling(Scrambling(test)) << "\n";  
 cout << "Check:\t\t";  
 check(test);  
 return 0;  
}

**Результат работы программы**



**Физическое кодирование**



**Достоинства и недостатки скремблирования**

|  |  |
| --- | --- |
| **Достоинства** | **Недостатки** |
| не уменьшается полезная пропускная способность канала связи, т. к. отсутствуют избыточные биты | дополнительные затраты в узлах сети на реализацию алгоритма скремблирования-дескремблирования |
| не всегда удается исключить длинные последовательности нулей и единиц |

**Вывод**

Использования сремблирования совместно с манчестерским кодом – излишество. Общая картина от данной операции не меняется в лучшую сторону. При этом появляются дополнительные затраты на сам алгоритм.

Скремблирование негативно сказывается на AMI и NRZI. Количество тактов с постоянной последовательность нулей и единиц стало больше.