



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» КЗ-МФ

Лабораторная работа №3

ПО ДИСЦИПЛИНЕ:

Сети ЭВМ и телекоммуникации

НА ТЕМУ:

Логическое кодирование:

скремблирование

Студент КЗ-53Б
(Группа)

(Подпись, дата)

Цветков Юрий Алексеевич
(И.О.Фамилия)

Преподаватель

(Подпись, дата)

Подворная Анастасия Витальевна
(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

$$B_i = A_i \oplus B_{(i-5)} \oplus B_{(i-9)} \oplus B_{(i-11)}$$

$$C_i = B_i \oplus B_{(i-5)} \oplus B_{(i-9)} \oplus B_{(i-11)}$$

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 = "110101100000000011011110111111111000000";
    cout << "Code:\t\t" << test << "\n";
    cout << "Scrambling:\t" << Scrambling(test) << "\n";
    printf("16bit res:\t%lX\n", strtoull(Scrambling(test).c_str(), NULL, 2) );
    cout << "deScrambling:\t" << deScrambling(Scrambling(test)) << "\n";
    cout << "Check:\t\t";
    check(test);
    return 0;
}

```

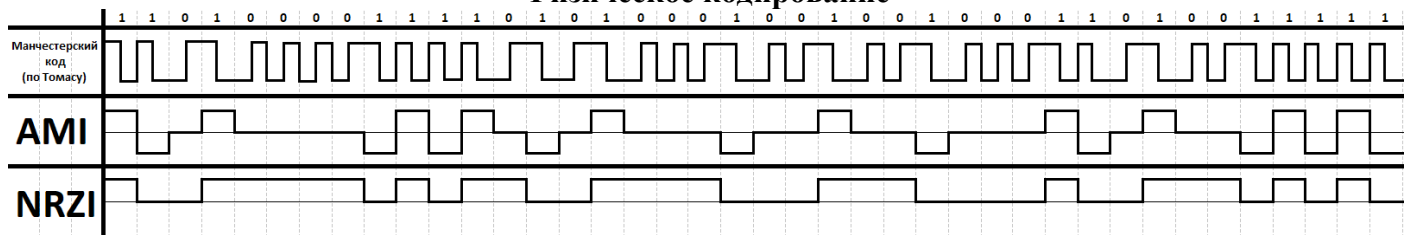
Результат работы программы

```

Run: logic_coding x
C:\Users\mr lux\Desktop\logic_coding\cmake-build-debug\logic_coding.exe
Code:          110101100000000011011110111111111000000
Scrambling:    1101000011110101000100100100011010011111
16bit res:     D0F512469F
deScrambling: 110101100000000011011110111111111000000
Check:         OK
Process finished with exit code 0

```

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 удастся исключить длинные последовательности нулей и единиц