**Εργασία Ψηφιακές Επικοινωνίες**

Γεώργιος Λαζαρίδης ΑΕΜ: 4419

Α)

Το πρόγραμμα υπολογισμού CRC σε γλώσσα C++ (Εφαρμογή: VS Code)

#include <iostream>

#include <string>

#include <vector>

#include <cstdlib>

#include <ctime>

using namespace std;

string generateMessage(int k) { //generation of random message

    string message;

    for (int i = 0; i < k; ++i) {

        message += rand() % 2 ? '1' : '0';

    }

    return message;

}

string xorStrings(const string &a, const string &b) { //xor function for the mod2 division

    string result;

    for (size\_t i = 1; i < b.size(); ++i) {

        result += (a[i] == b[i]) ? '0' : '1';

    }

    return result;

}

string mod2div(string dividend, const string &divisor) { //mod2 division

    int pick = divisor.size();

    string tmp = dividend.substr(0, pick);

    while (pick < dividend.size()) {

        if (tmp[0] == '1') {

            tmp = xorStrings(divisor, tmp) + dividend[pick];

        } else {

            tmp = xorStrings(string(pick, '0'), tmp) + dividend[pick];

        }

        ++pick;

    }

    if (tmp[0] == '1') {

        tmp = xorStrings(divisor, tmp);

    } else {

        tmp = xorStrings(string(pick, '0'), tmp);

    }

    return tmp;

}

string calculateCRC(const string &message, const string &P) { //calculation of CRC

    string message\_with\_0 = message + string(P.size() - 1, '0');

    return mod2div(message\_with\_0, P);

}

string transmitMessage(const string &message, const string &crc, double ber) { //transmitting message through a noisy channel

    string transmitted\_message = message + crc;

    string noisy\_message;

    for (char bit : transmitted\_message) {

        if ((double)rand() / RAND\_MAX < ber) {

            noisy\_message += (bit == '1') ? '0' : '1';

        } else {

            noisy\_message += bit;

        }

    }

    return noisy\_message;

}

bool checkCRC(const string &received\_message, const string &divisor) { //checks crc at receiver end

    return mod2div(received\_message, divisor).find('1') == string::npos;

}

int main() {

    srand(time(0)); //used for the random message generator

    int k = rand() % 15; //creation of random k vat for message length. upper bound 15 for simplicity, works without too.

    string P;

    cout << "Δώστε δυαδικό αριθμό για πρότυπο υπολογισμού του CRC:" << endl;

    cin >> P; //the divisor used for the CRC

    double BER = 10^-3;

    string message = generateMessage(k); //generating the random message with max length k

    cout << "Αρχικό μήνυμα: " << message << endl;

    string crc = calculateCRC(message, P); //calculating the CRC of the message

    cout << "Υπολογισμένο CRC: " << crc << endl;

    string noisy\_message = transmitMessage(message, crc, BER); //passing the message through a noisy channel (with BER=0.01)

    cout << "Ληφθέν μήνυμα: " << noisy\_message << endl;

    bool is\_valid = checkCRC(noisy\_message, P); //checking if the message is valid

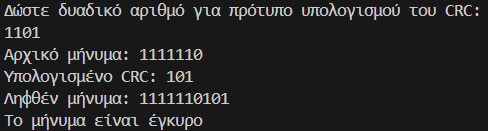
    cout << "Το μήνυμα είναι " << (is\_valid ? "έγκυρο" : "μη έγκυρο") << endl;

    return 0;

}

Παράδειγμα λειτουργίας του προγράμματος:

1.



2.

A screen shot of a computer

Description automatically generated

Επεξήγηση προγράμματος:

Το πρόγραμμα αξιοποιεί διάφορες βιβλιοθήκες. Η παραγωγή του τυχαίου μηνύματος γίνεται με την χρήση της rand() για το μέγεθος του μηνύματος στην οποία τοποθετείτε ανώτατο όριο 15 bits για λόγους κατανόησης. Το Ρ δίνεται από τον χρήστη και στο ΒΕR έχει δωθεί τιμή 10-3

Ο υπολογισμός του CRC γίνεται με την χρήση τριών συναρτήσεων:

1. calculateCRC: Προσθέτει στο μήνυμα Ρ-1 μηδενικά στο τέλος και τα αποθηκεύει
2. mod2div: Κάνει την modulo 2 διαίρεση μεταξύ του Ρ και του τροποποιημένου μηνύματος
3. xorStrings: Συνάρτηση υλοποίησης την πράξης XOR που χρησιμοποιείται στην modulo 2 διαίρεση.

Τέλος το μήνυμα «μεταδίδεται» μέσα από ένα θορυβόδες κανάλι με πιθανότητα σφάλματος BER στην συνάρτηση transmitMessage και ελέγχεται άμα τελικά το μεταδιδόμενο μήνυμα που λήφθηκε στην μεριά του παραλήπτη είναι σωστό.

Β) Για k=20, Ρ=110101 και BER=10-3

* Το ποσοστό των μηνυμάτων που φθάνουν με σφάλμα (στο block δεδομένων ή στο CRC) στον αποδέκτη. = 2.3%
* Το ποσοστό των μηνυμάτων που ανιχνεύονται ως εσφαλμένα από το CRC. = 2.2862%
* Tο ποσοστό των μηνυμάτων που φθάνουν με σφάλμα στο αποδέκτη και **ανιχνεύονται** από το CRC.= 99,4%
* Tο ποσοστό των μηνυμάτων που φθάνουν με σφάλμα στο αποδέκτη και **δεν ανιχνεύονται** από το CRC.= 0.6%