

Προγραμματιστική εργασία στο μάθημα «Ψηφιακές επικοινωνίες»

Φοιτητής: Ολγκέρ Χότζα
Τμήμα: Πληροφορική, Α.Π.Θ

ΠΕΡΙΕΧΟΜΕΝΑ

1	Παραδείγματα λειτουργίας	1
2	Σύντομη περιγραφή σημείων στο κώδικα	2
3	Λεπτομέρειές.....	5

1 ΠΑΡΑΔΕΙΓΜΑΤΑ ΛΕΙΤΟΥΡΓΙΑΣ

Στους παρακάτω πίνακες (βλ. Πίνακας 1.1, Πίνακας 1.2, Πίνακας 1.3, Πίνακας 1.4) παρατηρούμε τα αποτελέσματα μερικών παραδειγμάτων.

Όπου βλέπεται **N** είναι οι επαναλήψεις που θα κάνει (ο αριθμός μηνυμάτων), ενώ **K** το μήκος των ακολουθιών που θα παραχθούν και τέλος το **P** είναι ο προκαθορισμένος αριθμός των $n-k+1$ bits.

Δεδομένα		Αποτελέσματα		
K	20	Σφάλμα	Πλήθος	Ποσοστό
P	110101	Συνολικά σφάλματα	246374	{κενό}
BER	10^{-3}	Μη εντοπισμένα σφάλματα	116	0.047
N	10 000 000	Εντοπισμένα σφάλματα	246258	99.953

Πίνακας 1.1 – Παράδειγμα πρώτο.

Δεδομένα		Αποτελέσματα		
K	100	Σφάλμα	Πλήθος	Ποσοστό
P	110101	Συνολικά σφάλματα	1044852	{κενό}
BER	10^{-4}	Μη εντοπισμένα σφάλματα	343	0.033
N	100 000 000	Εντοπισμένα σφάλματα	1044509	99.967

Πίνακας 1.2 – Παράδειγμα δεύτερο.

Δεδομένα		Αποτελέσματα		
K	1000	Σφάλμα	Πλήθος	Ποσοστό
P	1100100100001011101	Συνολικά σφάλματα	10309	{κενό}
BER	10^{-5}	Μη εντοπισμένα σφάλματα	0	0.000
N	1 000 000	Εντοπισμένα σφάλματα	10309	100.000

Πίνακας 1.3 – Παράδειγμα τρίτο.

Δεδομένα		Αποτελέσματα		
K	10000	Σφάλμα	Πλήθος	Ποσοστό
P	1001101001	Συνολικά σφάλματα	99996	{κενό}
BER	10^{-3}	Μη εντοπισμένα σφάλματα	200	0.200
N	100 000	Εντοπισμένα σφάλματα	99796	99.800

Πίνακας 1.4 – Παράδειγμα τέταρτο.

2 ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ ΣΗΜΕΙΩΝ ΣΤΟ ΚΩΔΙΚΑ

Στα παρακάτω μπλοκ κώδικα ερμηνεύουμε κάποια σημαντικά σημεία του κώδικα. Θα υπάρχουν οι μέθοδοι αυτούσιες με κώδικα και σε σχόλιο (π.χ. `// (3)`) θα αντιστοιχεί ένας αριθμός όπου θα υπάρχει μια αντίστοιχη λίστα με τους αριθμούς αυτούς και θα σχολιάζουν το συγκεκριμένη χωρίο.

Η παρακάτω λίστα αναφέρετε στην συνάρτηση **Div** (βλ. Κώδικας 2.1) όπου πραγματοποιείται ο αλγόριθμος της δυαδικής διαίρεσης μεταξύ του T και του P. Παίρνει ως είσοδο την T Ακολουθία των n bits και τον P προκαθορισμένο αριθμός των n-k+1 bits. Επιστρέφει το αποτέλεσμα της διαίρεσης σε συμβολοσειρά.

- (1) Κάνουμε XOR το FCS με το P.
- (2) Αφαιρούμε όλα τα μηδενικά.
- (3) Αν δεν υπάρχει κανένα μηδενικό, τότε "αδειάζουμε" την FCS.
- (4) Αντιγράφοντας bits από το T στον FCS έως ότου το P και το FCS να έχουν ίδιο μήκος bits.
- (5) Τερματίζουμε τη διαδικασία όταν δεν υπάρχουν άλλα bits στο T για να αντιγράψουμε.
- (6) Συμπληρώνουμε μηδενικά bits στην περίπτωση που χρειάζεται bits έπειτα από το τελευταίο bit του T.

```
private String Div(String T, String P)
{
    int pointerOfT = P.length();
    this.FCS = T.substring(0, P.length());

    while (true)
    {
        // (1)
        this.FCS = XOR(this.FCS, P);
    }
}
```

```

    try
    {
        // (2)
        this.FCS = this.FCS.substring(this.FCS.indexOf("1"));

    } catch (Exception ignored)
    {
        // (3)
        this.FCS = "";
    }

    try
    {
        // (4)
        while (this.FCS.length() < P.length())
            this.FCS =
this.FCS.concat(Character.toString(T.charAt(pointerOfT++)));

    } catch (Exception ignored)
    {
        // (5)
        break;
    }
}

// (6)
while (this.FCS.length() < P.length() - 1)
    this.FCS = this.FCS.concat("0");

return this.FCS;
}

```

Κώδικας 2.1 – Μέθοδος DIV.

Από την άλλη, η μέθοδος **RUN** υλοποιεί τον αλγόριθμο CRC (βλ. Κώδικας 2.2) με την χρήση βοηθητικών μεθόδων όπως είναι η RANDOMBITGENERATOR και DIV. Δεν δέχεται κάποιο όρισμα και δεν επιστρέφει κάποια τιμή. Αποτελεί τη βασική συνάρτηση για να αρχινήσει η διαδικασία.

- (1) Παίρνουμε τιμές από τον χρήστη.
- (2) Αν είναι 0 ή 1, τότε προσθέτουμε το bit στο transData.
- (3) Διαδικασία ισότιμης πιθανότητας ελέγχου για το περιστατικό σφάλμα όπου αντιστρέφει τα bits που έχουν παραλάβει και τα επιστρέφει.
- (4) Αυξάνουμε το σύνολο με τα εντοπισμένα σφάλματα.
- (5) Ελέγχουμε αν το σφάλμα δεν έχει εντοπιστεί και αυξάνουμε τον αντίστοιχο μετρητή

```

public void Run()
{

```

```

// (1)
GetInput();

long loop = 0;
while (loop++ < N)
{
    String randomBits = RandomBitGenerator(K);
    String FCS = Div(randomBits.concat("0".repeat(P.length() - 1)), P);

    String received = randomBits.concat(FCS);
    StringBuilder transData = new StringBuilder();

    for (int i = 0; i < received.length(); i++)
    {
        // (2)
        if (new Random().nextDouble() > BER)
        {
            transData.append(received.charAt(i));
            continue;
        }

        // (3)
        transData.append(received.charAt(i) == '0' ? "1" : "0");
    }

    if (!received.equals(transData.toString()))
    {
        // (4)
        foundErrors++;

        // (5)
        if (Div(transData.toString(), P).equals("0".repeat(P.length() - 1)))
            notFoundErrors++;
    }
}
}

```

Κώδικας 2.2 – Μέθοδος RUN.

Οι υπόλοιπες μέθοδοι όπως είναι η **GETINPUT** καλείται από την RUN και χρησιμοποιείται για να εισάγει ο χρήστης τα αντίστοιχα δεδομένα. Περιορισμός τίθεται στα inputs του χρήστη στο P όπου θα πρέπει να είναι 0 ή 1 (και αυτό ελέγχεται με REGEX θέτοντας όριο τουλάχιστον ένα bit) και να ξεκινά και να τελειώνει σε άσο.

Αντίστοιχα η μέθοδος **PRINT** καλείται από την MAIN μέθοδο της κλάσης για να εμφανίσει στην οθόνη τα αποτελέσματα μετά από την διαδικασία του αλγορίθμου.

Μέθοδος που δημιουργεί τυχαιοποιημένες ακολουθίες bit με ίση πιθανότητα άσσου ή μηδέν αποτελεί η **RANDOMBITGENERATOR** που δέχεται ως όρισμα τον αριθμό K που δηλώνει το μήκος της ακολουθίας bit. Επιστρέφει την τυχαιοποιημένη ακολουθία. Δεν υπάρχει κάτι ιδιαίτερο για σχολιασμό σε αυτή τη μέθοδο.

Τέλος, η μέθοδος **XOR** υλοποιεί τον αλγόριθμο της διαίρεσης modulo-2 του FCS και του P. Δέχεται μια ακολουθία ελέγχου σφάλματος F των n-k bits, τον FCS και έναν προκαθορισμένο αριθμό των n-k+1 bits, τον P. Επιστρέφει το αποτέλεσμα της modulo-2 διαίρεσης.

3 ΛΕΠΤΟΜΕΡΕΙΕΣ

Σε περίπτωση που κάτι έχει πάει στραβά ή το πρόγραμμα δεν τρέχει, παρακάτω θα βρείτε την έκδοση JDK που έχω εγκατεστημένη και έχω επιβεβαιώσει ότι τρέχει.

```
>> java --version

openjdk 11.0.12 2021-07-20
OpenJDK Runtime Environment Microsoft-25199 (build 11.0.12+7)
OpenJDK 64-Bit Server VM Microsoft-25199 (build 11.0.12+7, mixed mode)
```