

# Εργασία στις Ψηφιακές Επικοινωνίες

Γραμμένος Θεόδωρος

A.E.M.: 3294

## 1 Πληροφορίες για την υλοποίηση

Η υλοποίηση έχει γίνει στη γλώσσα προγραμματισμού C++. Σε κάποια σημεία της εφαρμογής οι ακολουθίες από bit αποθηκεύονται ως unsigned integers (για παράδειγμα το  $P_2(110101) = P_{10}(53)$ ) και οι πράξεις γίνονται με πράξεις XOR πάνω στα bit των αριθμών.

Δημιουργήθηκε μία κλάση CRCManager η οποία περιέχει 4 στατικές μεθόδους που υλοποιούν το πρόβλημα. Οι παράμετροι του προβλήματος: n, k, BER και P δηλώνονται ως σταθερές μέσω DEFINE στο αρχείο src/CRCManager.cpp.

- Η μέθοδος createMessage δημιουργεί έναν τυχαίο μήνυμα k bit. Για να το κάνει αυτό επιλέγει από μία γεννήτρια παραγωγής τυχαίων αριθμών έναν αριθμό από το 0 έως το  $2^k - 1$ . Έτσι καλύπτεται κάθε ακολουθία k bit.
- Η μέθοδος addFCS δέχεται ως όρισμα τα δεδομένα για τα οποία θα υπολογίσει το CRC. Αφού υπολογίσει το CRC το τοποθετεί δεξιά από τα δεδομένα και επιστρέφει το μήνυμα που θα μεταδωθεί.
- Η μέθοδος addErrors δέχεται ως όρισμα το μήνυμα που θα μεταδοθεί. Έπειτα χρησιμοποιεί μια γεννήτρια τυχαίων αριθμών, βασισμένη στην κατανομή Bernoulli με  $p = 10^{-3}$  για να αποφασίσει αν θα αλλάξει ένα bit, ή όχι. Όταν η γεννήτρια επιστρέψει true, αλλάζει το συγκεκριμένο bit. Η συνάρτηση επιστρέφει true εάν άλλαξε ένα ή περισσότερα bit στο μήνυμα.
- Η μέθοδος verify δέχεται ως όρισμα το μήνυμα και επιστρέφει εάν το μήνυμα είναι ορθό, σύμφωνα με το CRC.

Στη μετάδοση ενός μηνύματος οι μέθοδοι `createMessage` και `addFCS` προσομοιώνουν τη διαδικασία που γίνεται στον αποστολέα του μηνύματος, η μέθοδος `addError` προσομοιώνει τη μετάδοση του μηνύματος και η `verify` τη διαδικασία που γίνεται στον αποδέκτη.

Η συνάρτηση `main` δημιουργεί τον αριθμό μηνυμάτων που ορίζεται στο `MESSAGE_NUMBER` και υπολογίζει: πόσα μηνύματα μεταδόθηκαν χωρίς σφάλμα, πόσα μεταδόθηκαν με σφάλμα και αυτό εντοπίστηκε από το CRC και πόσα μεταδόθηκαν με σφάλμα που δεν εντοπίστηκε από το CRC. Προκειμένου αυτό να γίνει πιο γρήγορα χρησιμοποιείται `multi-threading`, ανάλογα με τις δυνατότητες του συστήματος.

## 1.1 Παράδειγμα

Για  $k = 12, n = 20, BER = 0.1, P = 100011101$  εκτελείται η μέθοδος `createMessage` και επιστρέφει τον αριθμό  $(2387)_{10}$  που αντιστοιχεί στην ακολουθία bit  $(101100010101)_2$ .

Στη συνέχεια, εκτελείται η μέθοδος `addFCS` και επιστρέφει τον αριθμό 726314 που αντιστοιχεί στα bit 10110001010100101010. Επιβεβαιώνεται ότι τα πρώτα 12 bit αντιστοιχούν στο μήνυμα.

Έπειτα εκτελείται η μέθοδος `addError` και αλλάζει το μήνυμα σε:

11111001010100101010. Στο μήνυμα έχουν αλλάξει αρκετά bit, κάτι που είναι αναμενόμενο λόγω του σχετικά υψηλού BER. Τέλος, η μέθοδος `verify` επιστρέφει `false`, καθώς το μήνυμα έχει αλλοιωθεί.

## 2 Στατιστικά

Από τα δεδομένα της άσκησης έχουμε ότι  $k = 10, P = 110101$  και  $BER = 10^{-3}$ . Συνεπώς  $n - k + 1 = 6$  άρα  $n = 15$ . Το μέγεθος του μηνύματος είναι 15 bits με τα 10 να αποτελούν το μήνυμα και τα υπόλοιπα 5 το FCS.

Η συνάρτηση `main` για 5.000.000.000 μηνύματα παράγει τα εξής αποτελέσματα:

- Αποστέλλονται 5.000.000.000 μηνύματα.
- Από αυτά 4.925.540.295 αποστέλλονται χωρίς λάθη
- 74.459.705(0, 01%) αποστέλλονται με σφάλμα σε ένα ή περισσότερα bits
- Σε όλα τα μηνύματα το CRC εντοπίζει το σφάλμα.

Παρατηρούμε ότι ακόμα και με έναν αρκετά μεγάλο αριθμό μηνυμάτων το CRC καταφέρνει και εντοπίζει όλα τα σφάλματα. Αυτό μας δείχνει πως η πιθανότητα να αλλάξουν τα κατάλληλα bit ώστε να μην εντοπίζεται το σφάλμα, είναι υπερβολικά μικρή με το τρέχον BER.

(Σημείωση: Το ποσοστό που εμφανίζει το πρόγραμμα διαφέρει από το πραγματικό ποσοστό λόγω της περιορισμένης ακρίβειας των double αριθμών, επειδή χρησιμοποιούνται οι τύποι unsigned long long που καταλαμβάνουν 64-bit.)

```
Thread 1:625000000 messages tested(50%)
Thread 3:625000000 messages tested(50%)
Thread 4:937500000 messages tested(75%)
Thread 1:937500000 messages tested(75%)
Thread 2:937500000 messages tested(75%)
Thread 3:937500000 messages tested(75%)
Messages Tested: 5000000000
Normal Messages: 4925540295
Messages with errors: 74459705 1.34301e-06%
---Error caught: 74459705 1.34301e-06%
---Error not caught: 0 0%
```

Σχήμα 1: Η έξοδος του κώδικα