

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Μάθημα: Ψηφιακές επικοινωνίες.

Όνομα: Μουστάκας Αριστείδης

AEM: 2380

email: moustakas@csd.auth.gr

α) Για την αναπαράσταση του κώδικα CRC υλοποίησα τις παρακάτω λειτουργίες. Αρχικά έφτιαξα τρεις κλάσεις, την transmitter η οποία υλοποιεί την έννοια του αποστολέα του μηνύματος, την receiver που υλοποιεί την έννοια του δέκτη του μηνύματος και επίσης μια κλάση message η οποία αναπαριστά ένα μήνυμα(έχει μέσα ένα array από 0 και 1 όπου είναι το μήνυμα και μια λογική μεταβλητή που είναι θετική αν το μήνυμα είναι σωστό αλλιώς αν είναι αρνητική το μήνυμα έχει αλλοιωθεί) . Έφτιαξα ένα απλό μενού και ρωτάω τον χρήστη αν θέλει να τρέξει ένα έτοιμο παράδειγμα ή να τρέξει ένα παράδειγμα με δικά του νούμερα. Έπειτα στην κύρια κλάση μου(Project_CRC) μέσα στην οποία υπάρχει και η εκτελέσιμη συνάρτηση(main), δημιουργώ ένα αντικείμενο αποστολέα και ένα δέκτη. Στην συνέχεια καλώ την συνάρτηση sendMessage() του αποστολέα τόσες φορές όσες τα μηνύματα που θέλω να στείλω. Η συνάρτηση sendMessage() με την σειρά της δημιουργεί ένα τυχαίο σήμα και προσθέτει στο τέλος αυτού τόσα μηδενικά όσα είναι το μέγεθος του pattern που θα κάνει την κωδικοποίηση CRC πλην 1. Μετά καλείται η συνάρτηση modulo() που κάνει την διαίρεση του μηνύματος με το pattern και βρίσκει το υπόλοιπο αυτής. Επίσης η συνάρτηση modulo() βάζει το υπόλοιπο της διαίρεσης στην θέση των μηδενικών που είχαμε προσθέσει στο τέλος του μηνύματος. Έπειτα καλείται η συνάρτηση distortion() η οποία αλλοιώνει το σήμα μας ανάλογα με το bit-error-rate που έχουμε δώσει και επιστρέφει ένα αντικείμενο message στην sendMessage() η οποία με την σειρά της το επιστρέφει στην main. Η main έχοντας δεχθεί το μήνυμα που δημιούργησε ο αποστολέας, καλεί την συνάρτηση getMessage() του δέκτη η οποία δέχεται το μήνυμα και καλεί με την σειρά της την συνάρτηση modulo() που κάνει την διαίρεση του μηνύματος με το pattern. Αν αυτό βγει 0 τότε το μήνυμα μας έχει μεταδοθεί σωστά. Σε

αντίθετη περίπτωση, αν το υπόλοιπο δεν είναι 0(αυτό ελέγχεται μέσω της συνάρτησης `isCorrect()`) τότε το σήμα μας είναι αλλοιωμένο(υπό κανονικές συνθήκες ο δέκτης κάνει αίτηση στον αποστολέα να του ξαναστείλει το συγκεκριμένο μήνυμα). Τέλος ο δέκτης επιστρέφει στην `main` μια λογική μεταβλητή για το αν το σήμα είναι αλλοιωμένο (σύμφωνα με την διαίρεση που έχει κάνει, δηλαδή μπορεί να είναι στην πραγματικότητα λάθος αλλά να το βρήκε σωστό). Έτσι η `main` μας εκτυπώνει το ποσοστό των μηνυμάτων που είχαν υποστεί αλλοίωση σε σχέση με τον συνολικό αριθμό μηνυμάτων που στείλαμε, το ποσοστό των μηνυμάτων που είχαν λάθος και τα αναγνώρισε ο δέκτης σε σχέση με τον συνολικό αριθμό μηνυμάτων που είχαν λάθος και το ποσοστό των μηνυμάτων που είχαν αλλοιωθεί αλλά δεν τα αναγνώρισε ο δέκτης σε σχέση με τον συνολικό αριθμό μηνυμάτων που είχαν λάθος.

Μερικά παραδείγματα λειτουργίας:

- Example with number of messages= 1.000.000, Pattern= [1, 0, 0, 1, 1, 0, 0, 1], size of every message= 20 bits and ber=1/300
Messages with error 8.6163% of all messages.
Messages with error which were found is 99.940% of all messages with error.
Messages with error which were NOT found is 0.059% of all messages with error.
- Example with number of messages= 1.000.000, Pattern= [1, 0, 0, 1, 1, 0, 1], size of every message= 15 bits and ber=1/500.
Messages with error 4.1286% of all messages.
Messages with error which were found is 100.0% of all messages with error.
Messages with error which were NOT found is 0.0% of all messages with error.
- Example with number of messages= 10.000.000, Pattern= [1, 0, 0, 1, 0, 1], size of every message= 25 bits and ber=1/150.
Messages with error 18.190% of all messages.
Messages with error which were found is 99.979% of all messages with error.
Messages with error which were NOT found is 0.020% of all messages with error.
- Example with number of messages= 10.000.000, Pattern= [1, 0, 1, 0, 1], size of every message= 30 bits and ber=1/100.
Messages with error 28.95395% of all messages.
Messages with error which were found is 97.8962% of all messages with error.

Messages with error which were NOT found is 2.1037% of all messages with error.

β) Αν $k=10$, $P=110101$ και $E=10^{-3}$.

Example with number of messages=1.000.000, Pattern={1,1,0,1,0,1}, size of every message=10 and ber=1/1000.

Messages with error 1.49374% of all messages.

Messages with error which were found is 100.0% of all messages with error.

Messages with error which were NOT found is 0.0% of all messages with error.

Για να περάσει ένα αλλοιωμένο μήνυμα από τον CRC χωρίς αυτός να το εντοπίσει πρέπει στον δέκτη, το υπόλοιπο της διαίρεσης του μηνύματος με το πρότυπο να είναι ίσο μηδέν. Για να συμβεί αυτό πρέπει να αλλοιωθούν περισσότερα από ένα bit σε ένα σήμα. Επομένως σε αυτό το παράδειγμα που έχουμε bit error rate ίσο με 10^{-3} και συνολικό μέγεθος μηνύματος(μαζί με τα bits ελέγχου) ίσο με 15 bits, είναι εξαιρετικά απίθανο να αλλοιωθούν στο ίδιο σήμα 2 bits. Αυτό είναι και ο λόγος που ο CRC εντοπίζει όλα τα μηνύματα που έχουν λάθη. Τέλος αξίζει να αναφέρουμε ότι σε περίπτωση όπου το ber ήταν πιο μεγάλο ή το σήμα μου ήταν πιο αποτελούταν από περισσότερα bits τότε θα υπήρχε μεγαλύτερη πιθανότητα να αλλοιωθούν περισσότερα από 1 bit σε κάποιο σήμα και τότε ίσως να μην το εντόπιζε ο CRC.