ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Εργασία CRC στο μάθημα "Ψηφιακές Επικοινωνίες"
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Τμήμα Πληροφορικής

Ονοματεπώνυμο : Ιωάννης Ντούβελος
ΑΕΜ:3340

Περίληψη : Η εργασία υλοποιήθηκε σε γλώσσα Java και περιλαμβάνει τις εξής λειτουργίες , δημιουργία τυχαίων μηνυμάτων με ισοπίθανα 0 και 1, επιστροφή της xor τιμής δύο ακεραίων , εύρεση του CRC που αντιστοιχεί σε κάθε μήνυμα βάσει του P αριθμού που δίνει ο χρήστης , αποστολή του μηνύματος μέσω καναλιού με συγκεκριμένο BER= 10^-3 όπως ορίζει η εκφώνηση , έλεγχος του CRC στην πλευρά του αποδέκτη και εμφάνιση στατιστικών για το ποσό των μηνυμάτων που είχαν λάθος , που ανιχνεύθηκαν ως λάθος από το CRC και που είχαν λάθος αλλά δεν ανιχνεύθηκαν.

Δημιουργία τυχαίων μηνυμάτων : Η συνάρτηση που υλοποιεί αυτή την λειτουργία είναι η void message_generator(int k) με όρισμα ακέραιο κ που δηλώνει πόσα ψηφία θέλουμε να έχει το μήνυμα . Χρησιμοποιώντας την Math.random() της Java η οποία επιστρέφει αριθμό από 0-1 , όταν βρίσκεται η τιμή που επιστρέφει πάνω από 0,5 τότε εισάγουμε στο μήνυμα 1 αλλιώς βάζουμε 0 . Το μήνυμα αποθηκεύεται στην Arraylist<Integer > message .

Επιστροφή της xor τιμής δύο ακεραίων : Η συνάρτηση που υλοποιεί αυτήν την λειτουργία είναι int xor_function(int a,int b) , επιστρέφει ακέραιο (την τιμή xor) και παίρνει σαν όρισμα δύο ακέραιους α και β , αν είναι ίσοι επιστρέφει 0 αλλιώς επιστρέφει 1 .

Εύρεση της τιμής CRC του μηνύματος βάσει του P αριθμού που δίνει ο χρήστης: Μόλις ξεκινάει το πρόγραμμα ζητείται από τον χρήστη να δώσει τον αριθμό P βάσει του οποίου θα γίνει η διαίρεση modulo 2 για να βρεθεί το CRC . Έπειτα , η συνάρτηση που υλοποιεί την διαίρεση είναι η void division(Arraylist<Integer> message , boolean check) η οποία παίρνει δύο ορίσματα ,το μήνυμα το οποίο πρέπει να διαιρέσει και μια λογική μεταβλητή . Και τα δύο αυτά ορίσματα δίνονται για να ξεχωρίσουμε σε ποια πλευρά βρισκόμαστε , πομπού ή δέκτη . Συγκεκριμένα , προσθέτουμε μηδενικά στο τέλος του μηνύματος μόνο όταν είμαστε στην πλευρά του πομπού , αντιγράφουμε το μήνυμα μόνο όταν είμαστε την πλευρά του πομπού (για να το κρατήσουμε ανέπαφο για τον έλεγχο των αποτελεσμάτων) και καλούμε την συνάρτηση για τον έλεγχο των αποτελεσμάτων (για τα στατιστικά) μόνο όταν βρισκόμαστε στην πλευρά του δέκτη. Εκτός από αυτές τις διαφορές στις δύο πλευρές η υπόλοιπη συνάρτηση παραμένει ίδια. Η υλοποίηση της διαίρεσης γίνεται με τον εξής τρόπο : Έχουμε ένα index το οποίο δείχνει κάθε χρονική στιγμή της εκτέλεσης ποιο ψηφίο του μηνύματος είναι το επόμενο για να “κατέβει” και έχουμε κι έναν πίνακα crc ο οποίος έχει μέγεθος ίδιο με του P αριθμού και κάθε χρονική στιγμή περιέχει τα αποτελέσματα από τις πράξεις xor που γίνονται . Αρχικά στον πίνακα εισάγονται τα αποτελέσματα της πρώτης πράξης xor του P με το μήνυμα . Στην συνέχεια αρχίζει η επανάληψη που τρέχει μέχρι να “κατέβει ” και το τελευταίο στοιχείο από το μήνυμα . Σε κάθε επανάληψη ελέγχουμε το πρώτο

στοιχείο του crc (αυτός είναι κι ο λόγος που την πρώτη πράξη την κάνουμε πριν ξεκινήσει η επανάληψη) , αν το στοιχείο είναι 1 κάνουμε κανονικά την πράξη xor (στοιχείο του μηνύματος με στοιχείο του P) και κρατάμε το αποτέλεσμα στον crc και συνεχίζουμε . Εάν το πρώτο στοιχείο είναι 0 τότε σημαίνει ότι πρέπει να το προσπεράσουμε και να κατεβάσουμε το επόμενο στοιχείο του μηνύματος . Αυτό γίνεται κάνοντας πράξη με xor 0 όλα τα στοιχεία εκτός από το πρώτο που είναι το 0 , και τα μετακινούμε μια θέση αριστερά στον πίνακα , στο τέλος κατεβάζουμε το στοιχείο που δείχνει το index στο μήνυμα και αυξάνουμε το index . Με αυτό τον τρόπο απαλείψαμε το μηδενικό από μπροστά και κατεβάσαμε το επόμενο στοιχείο χωρίς να επηρεαστεί κάποιο ενδιάμεσο ψηφίο αφού (0 xor 0 =0 και 1 xor 0 = 1) . Συνεχίζοντας έτσι όταν τελειώνει η επανάληψη ο πίνακας crc έχει μέσα 6 στοιχεία και έχει εξετάσει όλο το μήνυμα . Το CRC όμως πρέπει να είναι 5 ψηφία , οπότε για να πιάσουμε και την τελευταία περίπτωση που ξεφεύγει από την επανάληψη είναι να ελέγξουμε αν το πρώτο στοιχείο του crc είναι 1 , αν είναι τότε πρέπει να κάνουμε μια πράξη την οποία και κάνουμε κι έτσι στον πίνακα τα τελευταία 5 στοιχεία το αποτελούν το ζητούμενο CRC (τα τελευταία 5 από τα 6 διότι η τελευταία πράξη γίνεται μόνο σε περίπτωση που το πρώτο στοιχείο είναι 1 οπότε με τον πρώτο στοιχείο του P που είναι 1 θα προκύψει 0).

Αποστολή του μηνύματος μέσω καναλιού με BER : Η συνάρτηση που υλοποιεί αυτήν την λειτουργία είναι η void send_message() . Αρχικά προσθέτει το Crc που βρέθηκε από την διαίρεση στο τέλος του αρχικού μηνύματος (το αντίγραφο που είχαμε κρατήσει) , στην συνέχεια πραγματοποιεί την αποστολή στον δέκτη . Ο δέκτης στην ουσία είναι μια άλλη Arraylist<Integer> στη οποία μεταφέρουμε όλα τα στοιχεία του αρχικού μηνύματος ένα ένα χρησιμοποιώντας πάλι σαν πιθανότητα την Math.random()*1000 η οποία επιστρέφει έναν αριθμό από το 0 -1000 , άμα ο αριθμός είναι <1 τότε δημιουργούμε σφάλμα και στέλνουμε λάθος bit .

Έλεγχος Crc στην πλευρά του αποδέκτη : Ο έλεγχος υλοποιείται παλι με την συνάρτηση της διαίρεσης όπως προαναφέρθηκε μόνο με την διαφορά ότι καλούμε την συνάρτηση με την λογική τιμή που είναι όρισμα στην division ως true . Έτσι αφού γίνει η διαίρεση θα τρέξει και η συνάρτηση για έλεγχο των αποτελεσμάτων .

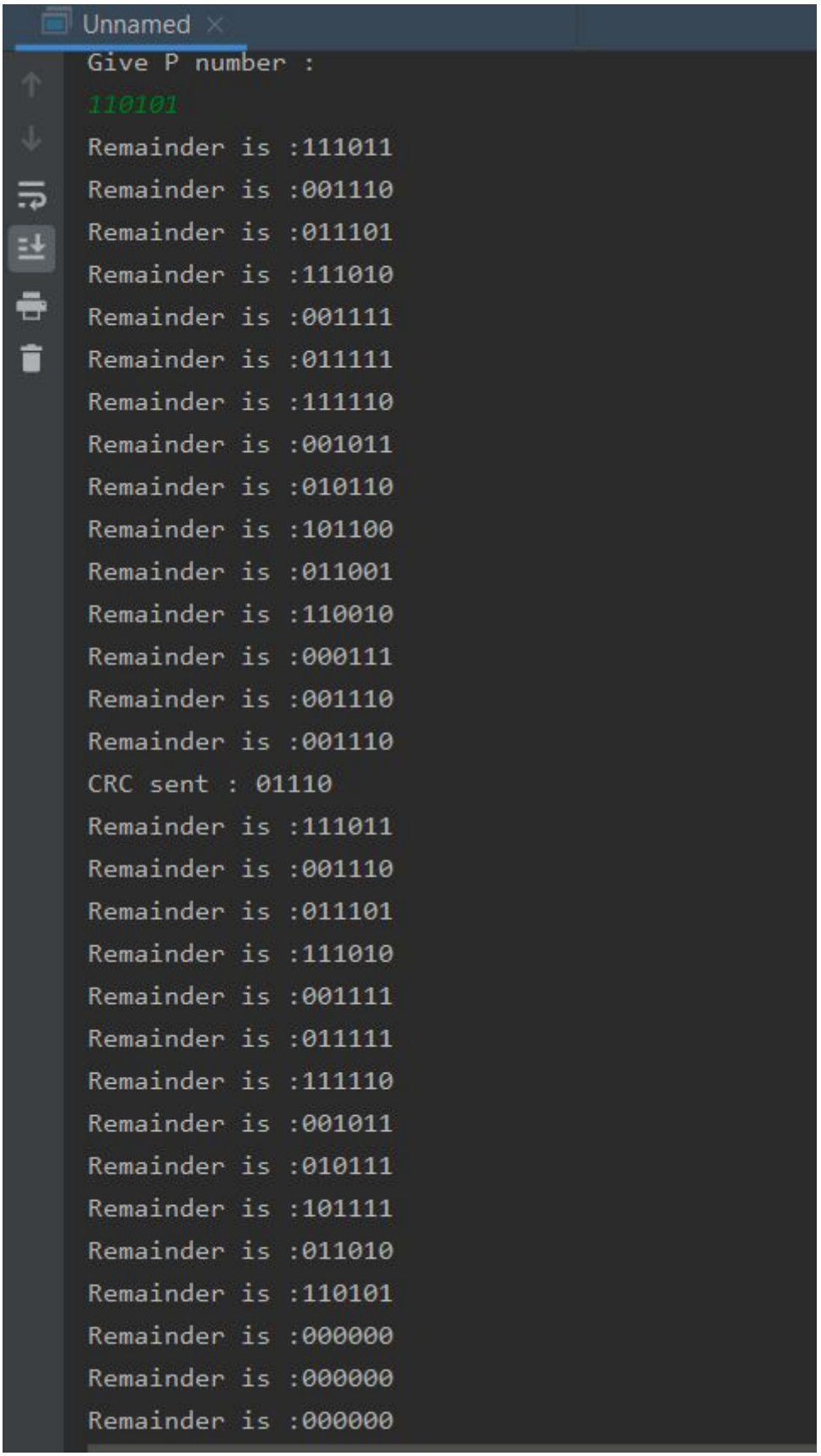
Δημιουργία στατιστικών : Η δημιουργία στατιστικών ξεκινάει από την συνάρτηση void check_result () η οποία καλείται κάθε φορά που το μήνυμα έφτασε στον αποδέκτη και έγινε έλεγχος για το Crc . Με τρεις ελέγχους βρίσκουμε αν υπήρξε πραγματικό σφάλμα στην αποστολή (έλεγχος ενα προς ενα τα στοιχεία που στάλθηκαν και έφτασαν) , αν ανιχνεύθηκε λάθος από το crc (αν μετά τον έλεγχο στον αποδέκτη ο πίνακας crc έχει μόνο μηδενικά σημαίνει ότι έφτασε χωρίς λάθος)

και τέλος ελέγχουμε ένα συνδυασμό των παραπάνω δύο , αν έφτασε μήνυμα με λάθος αλλά δεν ανιχνεύθηκε . Κρατάμε για την κάθε περίπτωση μία μεταβλητή και την αυξάνουμε κατά ένα κάθε φορά που βρούμε την αντίστοιχη περίπτωση . Στο τέλος εκτέλεσης του προγράμματος , στην main , αφού έχουμε ελέγξει έναν μεγάλο αριθμό μηνυμάτων , βάσει αυτών των μεταβλητών δημιουργούμε και εμφανίζουμε τα ποσοστά .

Συνάρτηση main () : Στην συνάρτηση main που τρέχει το πρόγραμμα ζητάμε από τον χρήστη να μας δώσει τον P αριθμό τον οποίο τον εισάγουμε σε έναν πίνακα . Στην συνέχεια έχουμε μια επανάληψη η οποία υλοποιεί τον εξής κύκλο : δημιουργία μηνύματος , εύρεση του CRC του , αποστολή στον δέκτη , έλεγχος στον δέκτη και έλεγχος αποτελεσμάτων και τέλος “καθαρίζουμε ” τις λίστες ώστε να περάσουμε στο επόμενο μήνυμα . Έτσι , δηλώνοντας σε μια μεταβλητή στην αρχή της κλάσης number_of_messages , απλά αλλάζουμε τον αριθμό της και έχουμε όσα μηνύματα θελήσουμε . Τέλος, δημιουργούνται και εμφανίζονται τα στατιστικά .

Σημείωση : Στον κώδικα θα αφεθούν σαν σχόλια κάποιες εντολές εμφάνισης για να ελεγχθούν αν χρειαστεί ακριβώς όλες οι πράξεις και τα αποτελέσματα από κάθε μήνυμα . Παραθέτονται screenshots από εκτέλεση για το παράδειγμα των διαφανειών και από εκτελέσεις για μεγάλο αριθμό μηνυμάτων , συγκεκριμένα για αριθμό χιλίων μηνυμάτων εώς 100 εκατομμυρίων .

Παράδειγμα εκτέλεσης για μήνυμα : 1010001101 και P : 110101



Στατιστικά για μεγάλο αριθμών μηνυμάτων :