

Ψηφιακές Επικοινωνίες - Εργαστηριακή Άσκηση 7

Έλεγχος ισοτιμίας– Parity Check

Γλώσσα Προγραμματισμού: Java (11)

Αρχείο υλοποίησης: “Lab7Exercise.java”

Ονοματεπώνυμο: Μπαρακλilής Ιωάννης

AEM: 3685

email: imparaki@csd.auth.gr

(Σημείωση: κατά την μεταγλώττιση, σε περίπτωση προβλημάτων με κωδικοποίηση μπορεί να προστεθεί το “-encoding UTF-8” στην εντολή (γραμμής εντολών) για μεταγλώττιση. Πχ. να χρησιμοποιηθεί η εντολή “javac Lab7Exercise.java -encoding UTF-8” για μεταγλώττιση.)

Άσκηση 1.

Περιγραφή έννοιας ισοτιμίας

Η ισοτιμία είναι μία (απλή) μέθοδος ανίχνευσης σφαλμάτων μεταδιδόμενων μηνυμάτων σε επικοινωνιακά συστήματα. Ο τρόπος με τον οποίο επιτυγχάνεται αυτή είναι η προσθήκη ενός επιπλέον bit σε κάθε μεταδιδόμενο μήνυμα έτσι ώστε ο συνολικός αριθμός bit (του μηνύματος και του πρόσθετου αυτού bit) με τιμή 1 να είναι άρτιος (σε άρτια ισοτιμία) ή περιττός (σε περιττή ισοτιμία).

Σχολιασμός σχετικού κώδικα και αποτελεσμάτων

Ζητείται να δημιουργηθεί κώδικας που να υπολογίζει άρτια ισοτιμία για ένα μήνυμα των 14 bits και να προσαρτά σε αυτό ένα bit ισοτιμίας.

Αυτό μπορεί να πραγματοποιηθεί με την χρήση της μεθόδου `generateEvenParity`¹. Αυτή, δέχεται ένα μήνυμα (χωρίς περιορισμούς προς το μέγεθος οπότε λειτουργεί και για μηνύματα 14 bit) ως αλφαριθμητικό (String) όπου σε κάθε χαρακτήρα αντιστοιχεί ένα bit του μηνύματος. Στην συνέχεια μετράει τον αριθμό των bit με τιμή 1. Αν ο αριθμός αυτός είναι άρτιος (διαίρεται με το 2 με υπόλοιπο 0, δηλαδή πληρείται η άρτια ισοτιμία) τότε το bit ισοτιμίας είναι 0 ώστε με την προσθήκη του να μην αλλάξει ο συνολικός αριθμός bit με τιμή 1 (και να μείνει άρτιος). Αντίστοιχα, όταν ο αριθμός αυτός είναι περιττός (δεν διαίρεται με το 2 με υπόλοιπο 0, δηλαδή δεν πληρείται η άρτια ισοτιμία) τότε το bit ισοτιμίας είναι 1 ώστε με την προσθήκη του να αλλάξει ο συνολικός αριθμός bit με τιμή 1 (και να γίνει άρτιος). Τέλος, προσαρτάται το bit ισοτιμίας που υπολογίστηκε στο δοθέν μήνυμα και επιστρέφεται η συνολική ακολουθία bits.

Στην συνέχεια, ζητείται να εκτελεστεί ο (παραπάνω) κώδικας για το μήνυμα 10001101011000. Αυτό γίνεται στις γραμμές 239-251 της συνάρτησης `main`² όπου στην συνέχεια εμφανίζονται στην προκαθορισμένη έξοδο τα αποτελέσματα. Αν εκτελέσουμε το πρόγραμμα, θα δούμε πως το αποτέλεσμα είναι “100011010110000” που αντιστοιχεί σε bit ισοτιμίας 0.

¹ Μέθοδος `generateEvenParity(String)` του αρχείου `Lab7Exercise.java` (γραμμές 3-41), υλοποιημένη σε γλώσσα Java.

² Μέθοδος `main(String[])` του αρχείου `Lab7Exercise.java` (γραμμές 230-340), υλοποιημένη σε γλώσσα Java.

Άσκηση 2.

Σχολιασμός σχετικού κώδικα και αποτελεσμάτων

Ζητείται δημιουργία κώδικα που να ελέγχει αν υπάρχει σφάλμα σε ένα bit για το μήνυμα των 14 bits που συνοδεύεται από ένα bit άρτιας ισοτιμίας.

Αυτό μπορεί να πραγματοποιηθεί με την χρήση της μεθόδου `checkEvenParity`³. Αυτή, δέχεται ένα μήνυμα που ακολουθείται (τελευταίος χαρακτήρας/ψηφίο) από το bit ισοτιμίας (δεν υπάρχουν περιορισμοί προς το μέγεθος οπότε λειτουργεί και για μηνύματα 14 bit μαζί με bit ισοτιμίας) ως αλφαριθμητικό (String) όπου σε κάθε χαρακτήρα αντιστοιχεί ένα bit του μηνύματος. Στην συνέχεια μετράει τον αριθμό των bit με τιμή 1. Τέλος, επιστρέφεται η λογική τιμή που αντιστοιχεί στο αν το πλήρες (συνυπολογίζοντας το bit ισοτιμίας) μήνυμα πληρεί τον κανόνα της άρτιας ισοτιμίας (αριθμός bit με τιμή 1 άρτιος που ισοδυναμεί με το να διαιρείται ακριβώς με το 2).

Επίσης, ζητείται να εκτελεστεί ο (παραπάνω) κώδικας για έλεγχο άφιξης χωρίς σφάλματος του μηνύματος 10001101011000 συνοδευόμενο από το bit ισοτιμίας του (που είναι 0 εφόσον το μήνυμα έχει 6 (άρτιο) αριθμό bit 1) στις τρεις περιπτώσεις άφιξης (με κανένα, ένα ή δύο σφάλματα αντίστοιχα).

Αυτό γίνεται στις γραμμές 253-288 (στις γραμμές 268-273, 275-280 και 282-287 για τις επιπτώσεις i, ii και iii αντίστοιχα) της συνάρτησης `main`² όπου στην συνέχεια εμφανίζονται στην προκαθορισμένη έξοδο τα αποτελέσματα. Αν εκτελέσουμε το πρόγραμμα, θα δούμε πως το αποτέλεσμα είναι ότι δεν ανιχνεύονται σφάλμα ισοτιμίας στην περίπτωση i και iii αλλά ανιχνεύεται στην ii.

Άσκηση 3.

Σχολιασμός σχετικού κώδικα και αποτελεσμάτων

Ζητείται δημιουργία κώδικα που να διαιρεί ένα μήνυμα των 28 bits σε 4 γραμμές των 7 bits, να υπολογίζει τα bits άρτιας διαστάσεως ισοτιμίας για κάθε γραμμή και κάθε στήλη και να τα προσαρτά σε αυτές.

Αυτό μπορεί να γίνει με την μέθοδο `generateTwoDimensionalEvenParity`⁴. Αυτή, δέχεται ένα μήνυμα 28 bit ως αλφαριθμητικό (String) όπου σε κάθε χαρακτήρα αντιστοιχεί ένα bit του μηνύματος. Στην συνέχεια, χωρίζει (νοητικά) το μήνυμα σε τμήματα των 7 bit όπου αντιστοιχεί κάθε 7-δα bits σε μία γραμμή του δισδιάστατου πίνακα, ενώ παράλληλα μετράει τον αριθμό bits με τιμή 1 της γραμμής υπολογίζοντας το bit ισοτιμίας της γραμμής (με όμοιο τρόπο με τον οποίο υπολογίστηκε το bit ισοτιμίας ενός μηνύματος στην άσκηση 1) και θέτει ως τελευταίο στοιχείο της γραμμής (στην 8η στήλη) το bit αυτό. Ακολούθως, μετά την συμπλήρωση του πίνακα και υπολογισμό του bit (άρτιας) ισοτιμίας κάθε γραμμής, γίνεται διάσχιση του πίνακα κατά στήλες ώστε να υπολογιστεί το bit ισοτιμίας κάθε στήλης και το bit αυτό τοποθετείται στον πίνακα ως το τελευταίο στοιχείο (5η γραμμή της στήλης). Τέλος, διαπερνάται ο πίνακας κατά γραμμές από αριστερά προς τα δεξιά και από πάνω προς τα κάτω ώστε να μετατραπεί σε ακολουθία από (40) bits και η ακολουθία αυτή επιστρέφεται ως αλφαριθμητικό που κωδικοποιεί τα bit ως χαρακτήρες.

Επίσης, ζητείται να εκτελεστεί ο κώδικας για το μήνυμα του δοθέντος σχήματος που είναι το 1100111101110101110010101001.

3 Μέθοδος `generateEvenParity(String)` του αρχείου `Lab7Exercise.java` (γραμμές 43-76), υλοποιημένη σε γλώσσα Java.

4 Μέθοδος `generateTwoDimensionalEvenParity(String)` του αρχείου `Lab7Exercise.java` (γραμμές 78-151), υλοποιημένη σε γλώσσα Java.

Στην συνέχεια, ζητείται να εκτελεστεί ο (παραπάνω) κώδικας για το μήνυμα 10001101011000. Αυτό γίνεται στις γραμμές 291-304 της συνάρτησης `main`² όπου στην συνέχεια εμφανίζονται στην προκαθορισμένη έξοδο τα αποτελέσματα. Αν εκτελέσουμε το πρόγραμμα, θα δούμε πως το αποτέλεσμα είναι “11001111011011011001001010011010101” που αντιστοιχεί στα bit του παραγόμενου πίνακα όπου κάθε 8-δα bits αντιστοιχεί σε μία γραμμή του πίνακα (όπου έχει προσαρτημένα τα bit ισοτιμίας).

Άσκηση 4.

Σχολιασμός σχετικού κώδικα και αποτελεσμάτων

Ζητείται δημιουργία κώδικα που να ανιχνεύει σφάλματα στον πίνακα δύο διαστάσεων (αποτελούμενο από bits μηνύματος και bits ισοτιμίας) που παράγεται από τον κώδικα του 3ου σκέλους της άσκησης.

Αυτό μπορεί να υλοποιηθεί με την μέθοδο `checkTwoDimensionalEvenParity`⁵. Αυτή, δέχεται ένα μήνυμα 40 bit ως αλφαριθμητικό (String) όπου σε κάθε χαρακτήρα αντιστοιχεί ένα bit του πίνακα όπου κάθε ομάδα 8 bits αντιστοιχεί σε μία γραμμή του (με προσαρτημένα τα bit ισοτιμίας γραμμών και στηλών) πίνακα. Στην συνέχεια, χωρίζει (νοητικά) το μήνυμα σε τμήματα των 8 bit όπου αντιστοιχεί κάθε 8-δα bits σε μία γραμμή του δισδιάστατου πίνακα με σκοπό την ανασυγκρότηση του. Μετά, υπολογίζονται τα bit άρτιας ισοτιμίας κάθε γραμμής και στήλης (των 7-δων αντί για 8-δων που λήφθηκαν) με τον ίδιο τρόπο με τον οποίο υπολογίστηκαν στην άσκηση 3 και μετά από κάθε υπολογισμό γίνεται έλεγχος για το αν το παραγόμενο bit ισοτιμίας ταυτίζεται με το bit ισοτιμίας που λήφθηκε. Αν σε κάθε έναν έλεγχο (κάθε στήλης και κάθε γραμμής) δεν εντοπιστεί κάποιο σφάλμα ισοτιμίας επιστρέφεται true. Διαφορετικά, αν δηλαδή εντοπιστεί έστω και ένα σφάλμα ισοτιμίας, επιστρέφεται false.

Επίσης, ζητείται να εκτελεστεί ο (παραπάνω) κώδικας για έλεγχο άφιξης χωρίς σφάλματος του μηνύματος που παράγεται από τον κώδικα του 3ου σκέλους της άσκησης (11001111011011011001001010011010101) στις τρεις περιπτώσεις άφιξης (με κανένα, ένα ή δύο σφάλματα αντίστοιχα).

Αυτό γίνεται στις γραμμές 317-338 (στις γραμμές 320-324, 326-331 και 333-338 για τις επιπτώσεις i, ii και iii αντίστοιχα) της συνάρτησης `main`² όπου στην συνέχεια εμφανίζονται στην προκαθορισμένη έξοδο τα αποτελέσματα. Αν εκτελέσουμε το πρόγραμμα, θα δούμε πως το αποτέλεσμα είναι ότι δεν ανιχνεύονται σφάλμα ισοτιμίας στην περίπτωση i, αλλά ανιχνεύεται στην ii και iii.

Άσκηση 5.

Απαντήσεις σε ερωτήματα: Υπάρχουν κάποια πρότυπα σφάλματος που δεν ανιχνεύονται στη δισδιάστατη ισοτιμία; Αν ναι, ποια είναι αυτά;

Ναι, υπάρχουν πρότυπα σφάλματος που δεν ανιχνεύονται στην δισδιάστατη ισοτιμία.

Πρότυπα σφάλματος που δεν ανιχνεύονται στην δισδιάστατη ισοτιμία είναι εκείνα με 4 σφάλματα bit σε θέσεις οι οποίες αποτελούν “γωνίες” ορθογωνίου (δηλαδή σχηματίζουν ορθογώνιο) στον δισδιάστατο πίνακα.

⁵ Μέθοδος `checkTwoDimensionalEvenParity(String)` του αρχείου `Lab7Exercise.java` (γραμμές 153-228), υλοποιημένη σε γλώσσα Java.