



Πανεπιστήμιο Θεσσαλίας

Διπλωματική Εργασία

Μελέτη και Βελτιστοποίηση Κωδίκων
Διαμόρφωσης με Περιορισμό Μήκους ως προς
την Ανθεκτικότητα στα Σφάλματα

Κάβουρας Στυλιανός
ΑΜ: 713

Επιβλέπων:

Σωτηρίου Χρήστος

Αργυρίου Αντώνιος

Επόπτης καθηγητής:

Κατσαβουνίδης Ιωάννης

Βόλος 2015

Στην οικογένειά μου και τους φίλους μου

© 2015 Κάβουρας Στυλιανός. All Rights Reserved.

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στους επιβλέποντες της διπλωματικής μου εργασίας κ. Ιωάννη Κατσαβουνίδη και κ. Αντώνιο Αργυρίου για την άψογη συνεργασία μας καθώς και για τις εύστοχες υποδείξεις τους, οι οποίες σε συνδυασμό με την εμπειρία και τις γνώσεις τους, αποτέλεσαν καθοριστικό παράγοντα για την εκπόνηση της συγκεκριμένης διπλωματικής εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω τον κ. Χρήστο Σωτηρίου. Εκτιμώ το γεγονός πως δέχτηκε να συνεργαστούμε με τόσο σύντομη προειδοποίηση.

Τέλος, οφείλω ένα μεγάλο ευχαριστώ στην οικογένειά μου και στους φίλους μου για την πολύτιμη και αέναη συμπαράστασή τους καθ' όλη τη διάρκεια της ακαδημαϊκής μου πορείας.

Περιεχόμενα

1	Εισαγωγή	6
1.1	Στόχοι της εργασίας	7
2	Θεωρητικό Υπόβαθρο	8
2.1	Ακολουθίες με Περιορισμό Μήκους (d,k)	8
2.2	Τρέχον Ψηφιακό Άθροισμα	10
2.3	Διαμόρφωση/Αποδιαμόρφωση EFM (8 προς 14)	11
2.4	EFM plus	13
3	Διαδικασία Διαμόρφωσης και Αποδιαμόρφωσης	16
3.1	Διαχωρισμός Καταστάσεων	16
3.2	Νέο Κριτήριο Επιλογής Κωδικών Λέξεων	18
3.3	Μέθοδος Διαμόρφωσης	19
3.4	Μέθοδος Αποδιαμόρφωσης	21
3.5	Βελτίωση του Βιβλίου Κωδικών του Standard DVD	23

4	Θεωρητική ανάλυση	26
4.1	Θεωρητική Ανάλυση RDS	26
4.1.1	Μοντελοποίηση Διαδικασίας Κωδικοποίησης	26
4.1.2	Στατική Κατανομή και Μεταβλητότητα RDS	27
4.2	Θεωρητική Ανάλυση Ανθεκτικότητας σε Σφάλματα	28
4.2.1	Υπολογισμός Πιθανότητας Σφάλματος από Αλλοίωση ενός Bit	29
4.2.2	Υπολογισμός Πιθανότητας Σφάλματος από Αλλοίωση ενός ή δύο Bit	31
4.2.3	Συνολικό Υπολειπόμενο Σφάλμα	32
5	Αποτελέσματα	34
5.1	Αποτελέσματα Μεταβλητότητας RDS	34
5.2	Αποτελέσματα Ανθεκτικότητας στα Σφάλματα	35
6	Συμπεράσματα και Μελλοντική εργασία	39

Κεφάλαιο 1

Εισαγωγή

Ο οπτικός δίσκος εφευρέθηκε το 1958. Το 1968 ο Αμερικανός εφευρέτης James Russel εφηύρε τον οπτικό δίσκο CD(Compact Disk)[10]. Οι οπτικοί δίσκοι πήραν την ονομασία τους εξαιτίας του μηχανισμού ανάγνωσης τους που είναι το φως και έγιναν γνωστοί την δεκαετία του 1980. Ακόμη και σήμερα αποτελούν ένα αποτελεσματικό μέσο αποθήκευσης δεδομένων, με το CD να χρησιμοποιείται κυρίως για την αποθήκευση ήχου και οι μετεξελίξεις του CD-R και CD-ROM να έχουν δυνατότητα αποθήκευσης και άλλου τύπου δεδομένα στη ίδια πάντα χωρητικότητα των 700 MB. Το DVD δημιουργήθηκε το 1995 και με ικανότητα αποθήκευσης 4 GB, ενώ πλέον έχει φτάσει μέχρι και τα 17 GB με το DVD-18 (διπλής όψεως).

Για την εγγραφή δεδομένων σε έναν οπτικό δίσκο χρησιμοποιείται μία ακτίνα που δημιουργεί οπές πάνω στην επιφάνειά του. Οπότε έχουμε δύο καταστάσεις την ύπαρξη οπής (κοιλότητα) και την απουσία της (νησίδα). Κατά την ανάγνωση του οπτικού δίσκου μια ακτίνα φωτός δημιουργείται από μια καθοδική δίοδο laser και κατευθύνεται σε συγκεκριμένα σημεία πάνω στην επιφάνεια του δίσκου. Αφού προσκρούσει πάνω στην επιφάνεια του δίσκου CD-ROM ανακλάται και με την βοήθεια μια σειράς συλλεκτών, καθρέφτες και ειδικούς φακούς η ανακλώμενη ακτίνα κατευθύνεται σε έναν ανιχνευτή φωτεινής δέσμης (photo-detector). Η ανάγνωση δίσκων με πολύ μικρές κοιλότητες και νησίδες είναι αρκετά δύσκολη και μπορεί να οδηγήσει σε σφάλματα λόγω τις αδυναμίας της ακτίνας ανάγνωσης να διαχωρίσει το πότε ξεκινάει κάποια κοιλότητα και πότε τελειώνει. Από την άλλη οι πολύ μεγάλες κοιλότητες/νησίδες που είναι ουσιαστικά πολλές διαδοχικές κοιλότητες/νησίδες οδηγούν σε σήμα πολύ χαμηλής συχνότητας που μπορεί να παρεμβάλλει με το σήμα που διορθώνει την τροχιά της ακτίνας. Το μεγαλύτερο πρόβλημα είναι αυτό της ανάκτησης ρολογιού [12]. Το ρολόι είναι εκείνο που ουσιαστικά διαβάσει ο πλήθος των νησίδων και κοιλοτήτων και ένα λάθος του μπορεί να αποτρέψει

την ανάγνωση του δίσκου. Για να λειτουργήσει σωστά το ρολόι, πρέπει να υπάρχουν συχνές εναλλαγές κοιλοτήτων και νησίδων. Υπάρχουν κανόνες που προσδιορίζουν το πόσο συχνές πρέπει να είναι αυτές οι εναλλαγές. Ένα ακόμη μειονέκτημα των οπτικών δίσκων είναι η εύκολη φθορά τους που οδηγεί σε λανθασμένη ανάγνωση των δεδομένων του. Μέχρι σήμερα τα σφάλματα κατά την ανάγνωση του δίσκου διορθώνονταν χρησιμοποιώντας κώδικες διόρθωσης λαθών (error correction codes) όπως Cross Interval Reed Solomon Codes [11]. Η χρήση τέτοιων κωδικών έχει ως αποτέλεσμα την αύξηση των αναγκαίων bit για την αποδιαμόρφωση των δεδομένων ενός δίσκου.

1.1 Στόχοι της εργασίας

Στόχοι αυτής της διπλωματικής είναι η μελέτη των μεθόδων διαμόρφωσης πριν την εγγραφή σε οπτικό δίσκο, η μελέτη των μεθόδων αποδιαμόρφωσης πριν την ανάγνωση του και τελικά η δημιουργία ενός εργαλείου που έχοντας ως είσοδο ένα βιβλίο κωδικών (codebook) θα υπολογίζει τις τιμές της μεταβλητότητας του RDS και της ανθεκτικότητας στα σφάλματα. Ακόμα θα προσπαθήσουμε να βελτιστοποιήσουμε το κάθε βιβλίο ως προς την ανθεκτικότητα του στα σφάλματα. Με αυτόν τον τρόπο θα μειώσουμε τα επιπλέον bit που χρησιμοποιούνται για την διόρθωση των σφαλμάτων. Οι μέθοδοι αυτοί είναι κώδικες με περιορισμούς μήκους που έχουν όσο γίνεται μικρότερη DC συνιστώσα. Οι μέχρι τώρα γνωστές μέθοδοι διαμόρφωσης αποδιαμόρφωσης είναι ο EFM και EFM-plus που χρησιμοποιούνται αντίστοιχα στο CD και DVD.

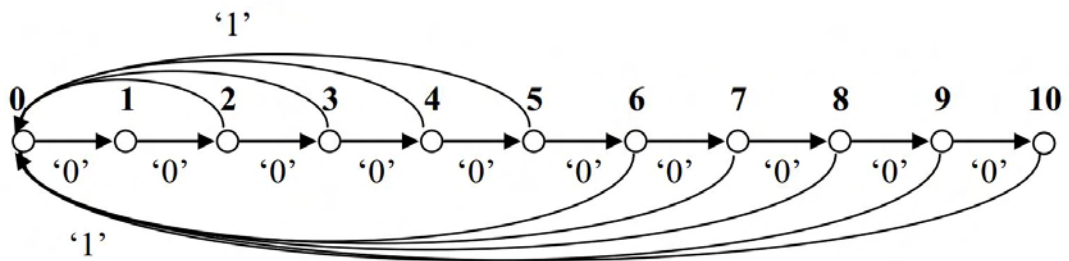
Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Ακολουθίες με Περιορισμό Μήκους (d,k)

Μία ακολουθία απο bit στην οποία μεταξύ δύο διαδοχικών άσων έχει από d έως το πολύ k μηδενικά ονομάζεται ακολουθία με περιορισμό μήκους (d,k)[1]. Ο περιορισμός αυτός βοηθάει στον να αποφύγουμε κοιλότητες ή νησίδες που θα έχουν αρκετά μεγάλο καθώς και αρκετά μικρό μέγεθος, διότι σε διαφορετική περίπτωση η ακτίνα ανάγνωσης του οπτικού δίσκου θα μπορούσε να οδηγηθεί σε κάποιο σφάλμα. Με την διαμόρφωση καναλιού μετατρέπονται τυχαίες ακολουθίες σε κώδικες που ικανοποιούν περιορισμούς που ορίζονται από τα χαρακτηριστικά του καναλιού, δηλαδή τους περιορισμούς μήκους (d,k). Οι κώδικες διαμόρφωσης που μελετάμε εμείς υπακούν στον περιορισμό μήκους (2,10) που σημαίνει ότι μεταξύ δύο άσων θα υπάρχουν τουλάχιστον 2 και το πού 10 μηδενικά αυτοί οι κώδικες ονομάζονται RLL(Run Length Limited).στο σχήμα 2.1 βλέπουμε τον γράφο που αναπαριστά τους κώδικες (2,10). Ο πίνακας γειτνίασης, δηλαδή ο πίνακας που ορίζει της έγκυρες μεταβάσεις μεταξύ δύο καταστάσεων του παραπάνω γράφου φαίνεται στο Σχήμα 2.2

Ο πολλαπλασιασμός του πίνακα με τον εαυτό του για 16 φορές μας δίνει τον αριθμό των ακολουθιών RLL που υπάρχουν για την μετάβαση από μία κατάσταση X σε μια Y . Για παράδειγμα στον Πίνακα 2.3 παρατηρούμε ότι για την μετάβαση από την κατάσταση 0 στην κατάσταση 3 υπάρχουν 27 διαφορετικές 16 bit ακολουθίες που πραγματοποιούν αυτή την μετάβαση.



Σχήμα 2.1: Γράφος ακολουθιών RLL [2]

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Σχήμα 2.2: Πίνακας γειτνίασης ακολουθιών (2,10)

$$A^{16} = \begin{pmatrix} 83 & 57 & 40 & 27 & 18 & 13 & 9 & 6 & 4 & 3 & 2 \\ 122 & 83 & 57 & 40 & 27 & 18 & 13 & 9 & 6 & 4 & 3 \\ 177 & 122 & 83 & 57 & 40 & 27 & 18 & 13 & 9 & 6 & 4 \\ 174 & 120 & 82 & 56 & 39 & 27 & 18 & 12 & 9 & 6 & 4 \\ 170 & 117 & 80 & 55 & 38 & 26 & 18 & 12 & 8 & 6 & 4 \\ 164 & 113 & 77 & 53 & 37 & 25 & 17 & 12 & 8 & 5 & 4 \\ 155 & 107 & 73 & 50 & 35 & 24 & 16 & 11 & 8 & 5 & 3 \\ 142 & 98 & 67 & 46 & 32 & 22 & 15 & 10 & 7 & 5 & 3 \\ 124 & 85 & 58 & 40 & 28 & 19 & 13 & 9 & 6 & 4 & 3 \\ 97 & 67 & 45 & 31 & 22 & 15 & 10 & 7 & 5 & 3 & 2 \\ 57 & 40 & 27 & 18 & 13 & 9 & 6 & 4 & 3 & 2 & 1 \end{pmatrix}$$

Σχήμα 2.3: Πίνακας μεταβάσεων του γράφου 2.1 για 16-bit ακολουθίες με περιορισμό μήκους (2,10)

Η χρήση τέτοιων ακολουθιών με περιορισμό μήκους βοηθάει στην επίλυση του προβλήματος ανάκτησης ρολογιού. Βέβαια για την διαμόρφωση ακολουθιών χαμηλής ισχύος αυτο δεν αρκεί.

2.2 Τρέχον Ψηφιακό Άθροισμα

Όπως αναφέραμε και στην παράγραφο 2.1, στην διαδικασία διαμόρφωσης μια ακολουθία ψηφίων μέσω κάποιων κωδικών διαμόρφωσης μετατρέπονται σε ακολουθίες RLL και είναι συνήθως μεγαλύτερου μήκους. Για την εγγραφή τους όμως στο οπτικό μέσω θα πρέπει να μετατραπούν σε μορφή NRZI (Non Return To Zero Inverted), δηλαδή σε ακολουθία από +1 και -1. Κάθε εναλλαγή πρόσημου υποδεικνύει την ύπαρξη ενός άσους στην αντίστοιχη RLL ακολουθία. Ως τρέχον ψηφιακό άθροισμα μιας NRZI ακολουθίας στη θέση i ορίζεται:

$$RDS_i = \sum_{j=0}^i NRZI_j$$

Παράδειγμα :

Έστω η ακολουθία $0x1008_{hex}$ δηλαδή 000100000001000_2 . Η NRZI αναπαράστασή της είναι:
 $+1 +1 +1 -1 -1 -1 -1 -1 -1 -1 -1 +1 +1 +1 +1$

Τα +1 και -1 υποδηλώνουν την ύπαρξη κοιλότητας ή νησίδας. Το ψηφιακό άθροισμα της παραπάνω ακολουθίας είναι -2.

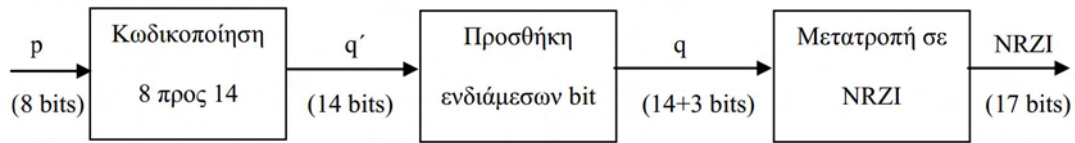
Βλέπουμε λοιπόν, ότι από μία (d,k) RLL ακολουθία προκύπτει μία NRZI ακολουθία με τουλάχιστον (d+1) και το πολύ (k+1) συνεχόμενα ίδια σύμβολα. Η παραπάνω NRZI ακολουθία μπορεί να γραφεί και ως η αντίθετη της ανάλογα με το πρόσημο που προηγούνταν του αρχικού μηδενικού μας, δηλαδή εάν είχαμε αρνητικό πρόσημο στα τελευταία ψηφία της προηγούμενης ακολουθίας τότε θα είχαμε $-1-1-1+1+1+1+1+1+1+1+1+1-1-1-1-1$.

Οι ακολουθίες που γράφονται στους οπτικούς δίσκους θα πρέπει να έχουν χαμηλή ισχύ στις χαμηλές συχνότητες διότι διαφορετικά θα υπάρχουν παρεμβολές με τη συχνότητα του σήματος διόρθωσης της τροχιάς της ακτίνας ανάγνωσης. Η μικρή ισχύς στο dc προϋποθέτει το πλήθος των κοιλοτήτων και νησίδων να είναι ίσο. Ακόμα ο περιορισμός μήκους που μας αποτρέπει να έχουμε περισσότερα από 10 μηδενικά μεταξύ δύο άσων εξασφαλίζει την συχνή εναλλαγή κοιλοτήτων και νησίδων το οποίο βοηθάει στην επίλυση του προβλήματος της ανάκτησης ρολογιού. Επίσης ο περιορισμός των 2 μηδενικών σχετίζεται με την ανάλυση του φακού στους οπτικούς δίσκους που είναι 3.

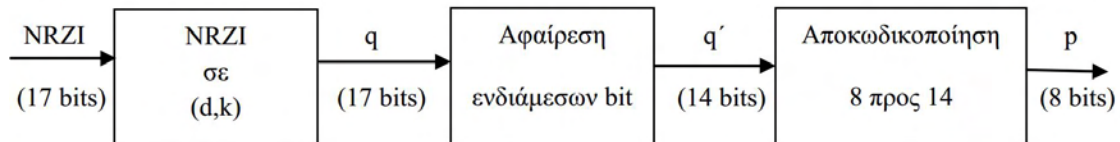
2.3 Διαμόρφωση/Αποδιαμόρφωση EFM (8 προς 14)

Η διαμόρφωση 8 προς 14, είναι η κωδικοποίηση που χρησιμοποιούν οι οπτικοί δίσκοι CD και αναπτύχθηκε από τους Immink και Ogawa στις αρχές τις δεκαετίας του 1980 [12]. Η διαμόρφωση 8 προς 14 παρουσιάζεται στο Σχήμα 2.4. Όπως βλέπουμε και στο σχήμα μετά την κωδικοποίηση έχουμε προσθήκη τριών επιπλέον bit έχοντας ως αποτέλεσμα την μείωση του ρυθμού κωδικοποίησης του αρχείου $\frac{8}{17}$. Ως ρυθμό κωδικοποίησης ορίζουμε τον λόγο των bit χρήστη ως προς το πλήθος των bit που γράφονται τελικά στον δίσκο.

Τα τρία επιπλέον bit που προστίθενται στην αποκωδικοποιημένη ακολουθία εξασφαλίζουν την τήρηση του περιορισμού μήκους (2,10) που είναι από της βασικές παραμέτρους της διαμόρφωσης EFM. Οι πιθανές τριάδες bit είναι 000, 001, 010 και 100. Η επιλογή της καταλληλότερης τριάδας γίνεται με βάση το πλήθος των αρχικών μηδενικών της



Σχήμα 2.4: Διαμόρφωση EFM



Σχήμα 2.5: Αποδιαμόρφωση EFM

αποδιαμορφωμένης κωδικής λέξης καθώς και το πλήθος των τελικών μηδενικών της προηγούμενης. Στην περίπτωση που περισσότερες από μία τριάδες μπορούν να χρησιμοποιηθούν, επιλέγεται εκείνη που οδηγεί σε χαμηλότερη ισχύ εγγραφής στις χαμηλές συχνότητες. Ουσιαστικά επιλέγεται η τριάδα που οδηγεί σε μικρότερο κατ' απόλυτη τιμή τρέχον ψηφιακό άθροισμα που ορίσαμε στο Κεφάλαιο 2.2.

Απουσία σφαλμάτων, η αποδιαμόρφωση 8 προς 14 μπορεί να γίνει όπως φαίνεται στο Σχήμα 2.5. Υποθέτοντας ότι η NRZI ακολουθία έχει ανεξάρτητο θόρυβο στα πρότυπα ενός δυαδικού συμμετρικού καναλιού με πιθανότητα σφάλματος ϵ , κάποια bit αντιστρέφονται και μπορεί η EFM κωδική λέξη που θα προκύψει (q') να μην είναι έγκυρη. Μια απλή λύση είναι να τερματίσει η αποκωδικοποίηση. Μπορούμε όμως να κάνουμε αποκωδικοποίηση μέγιστης πιθανοφάνειας, υπολογίζοντας για κάθε μη έγκυρη συμβολοσειρά την απόσταση Hamming (πλήθος διαφορετικών bit) από κάθε έγκυρη και υποθέτοντας ότι προέκυψε από αυτήν με την οποία μοιάζει περισσότερο (ελάχιστη απόσταση).

Άλλη μια τεχνική διαμόρφωσης αποδιαμόρφωσης είναι η 8 προς 14 16a (EFM 16a) η οποία σε αντίθεση με την παραπάνω χρησιμοποιεί 2 αντί για 3 ενδιάμεσα bit αυτό έχει ως αποτέλεσμα την αύξηση της χωρητικότητας του δίσκου διότι έχει πλέον ρυθμό κωδικοποίησης $\frac{8}{16}$, οπότε έχουμε $(8/16)/(8/17) = 1.0625$ δηλαδή περίπου 6 % περισσότερο χώρο. Η μέθοδος αυτή δεν χρησιμοποιείται σε πρακτικές εφαρμογές, ο λόγος είναι ότι αυξάνεται πολύ η ισχύς στις χαμηλές συχνότητες εξαιτίας των πολλών επιλογών για τα ενδιάμεσα bit.

2.4 EFM plus

Η διαμόρφωση 8 προς 14 συν (EFM plus) αναπτύχθηκε από τον Immink [3] και χρησιμοποιείται στους δίσκους DVD. Η βασική ιδέα που εισήγαγε ο Immink είναι η χρησιμοποίηση καταστάσεων τόσο στη διαμόρφωση όσο και στην αποδιαμόρφωση. Ο ρυθμός κωδικοποίησης είναι 8/16. Ο Shannon [4] απέδειξε ότι ο ρυθμός κωδικοποίησης δεν μπορεί να ξεπερνάει την χωρητικότητα Shannon και εξαρτάται αποκλειστικά από τους περιορισμούς μήκους. Η ισχύς στις χαμηλές συχνότητες μπορεί να πετύχει τιμές ανάλογες με της απλής διαμόρφωσης 8 προς 14 με ρυθμό κωδικοποίησης 8/17. Αυτό γίνεται με την εφαρμογή κατάλληλων τεχνικών, όπως η εναλλαγή καταστάσεων που θα εξηγήσουμε παρακάτω.

Ο Immink ύστερα από μία σειρά διαχωρισμών και συγχωνεύσεων κατέληξε σε βιβλία κωδίκων που αποτελούνται από τέσσερις καταστάσεις. Τα βιβλία αυτά είναι πρακτικά ένα είδος look-up table που χρησιμοποιεί ο κωδικοποιητής για να μετατρέψει της λέξεις εισόδου στις NRZI αναπαραστάσεις των ακολουθιών που αντιστοιχούν στην λέξη αυτή. Το ίδιο βιβλίο χρησιμοποιείται και για την αποκωδικοποίηση των NRZI στις λέξεις εισόδου που αντιστοιχούν (Σχήμα 2.6). Απόσπασμα του βιβλίου παρουσιάζεται στον Πίνακα 2.1 παρακάτω.

Insert code	State 1	Next state	State 2	Next state	State 3	Next state	State 4	Next state
0	0x2009	1	0x4024	2	0x2009	1	0x4024	2
1	0x2012	1	0x2012	1	0x8024	3	0x8024	3
2	0x2024	2	0x2024	2	0x8012	1	0x8012	1
3	0x2048	2	0x4480	4	0x2048	2	0x4480	4
4	0x2090	2	0x2090	2	0x8024	2	0x8024	2
5	0x2120	2	0x2120	2	0x8480	4	0x8480	4
6	0x2024	3	0x2024	3	0x8900	4	0x8900	4
7	0x2048	3	0x4012	1	0x2048	3	0x4012	1
8	0x2090	3	0x2090	3	0x9200	4	0x9200	4

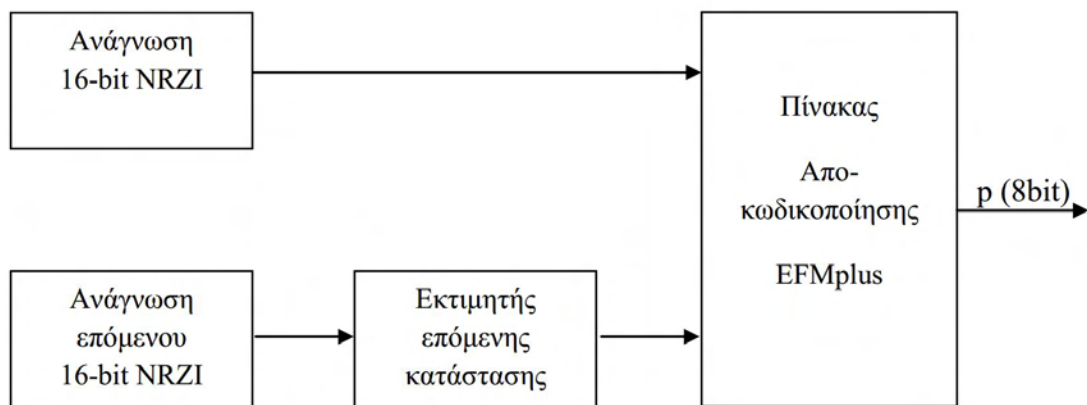
Πίνακας 2.1: Βιβλίο Κωδίκων τεσσάρων καταστάσεων

Ο διαχωρισμός των καταστάσεων έγινε με βάση το πλήθος των τελικών μηδενικών τις κάθε ακολουθίας RLL. Αναλυτικότερα οι κωδικές λέξεις που μεταβαίνουν στην κατά-σταση 1 έχουν m τελικά μηδενικά, $m \in [0, 1]$ ενώ οι κωδικές λέξεις που παράγονται έχουν n αρχικά μηδενικά, $n \in [2, 9]$. Οι κωδικές λέξεις που μεταβαίνουν στις καταστάσεις 2 και 3 έχουν m τελικά μηδενικά, $m \in [2, 5]$ ενώ οι κωδικές λέξεις που παράγονται έχουν n αρχικά μηδενικά, $n \in [0, 5]$. Τέλος Οι κωδικές λέξεις που μεταβαίνουν στην κατάσταση 4 έχουν m τελικά μηδενικά, $m \in [6, 9]$ ενώ οι κωδικές λέξεις που παράγονται έχουν n αρχικά μηδενικά $n \in [0, 1]$ [6] [5].

Μία ακόμα τεχνική που εφαρμόζει ο Immlink είναι αυτή της εναλλαγής καταστάσεων (state switching) από την κατάσταση 1 στην κατάσταση 4 και αντίστροφα. Όταν ο κωδικοποιητής βρίσκεται στην κατάσταση 1 έχει την δυνατότητα να επιλέξει μεταξύ της κωδικής λέξης σε αυτήν την κατάσταση και σε εκείνη της κατάστασης 4. Η επιλογή αυτή είναι εφικτή μόνο στην περίπτωση που δεν παραβιάζεται ο περιορισμός μήκους (d,k). Ο λόγος που επιτρέπουμε εναλλαγή μόνο μεταξύ των καταστάσεων 1 και 4 είναι ότι αν βρισκόμαστε σε μία από τις καταστάσεις 2 ή 3 τότε δεν πρέπει να κάνουμε εναλλαγή, για να εξασφαλίσουμε τη σωστή αποκωδικοποίηση. Η ένωση των καταστάσεων 2 και 3 είναι υποσύνολο των καταστάσεων 1 και 4 και λόγω της συνεπής ανάθεσης (consistent assignment) [2] αρκεί μόνο η εναλλαγή των καταστάσεων 1 και 4.

Αν παρατηρήσουμε ξανά τον Πίνακα 2.1 θα δούμε ότι στην κατάσταση ένα για παράδειγμα η λέξεις εισόδου 4 και 8 θα παράγουν την ίδια ακολουθία RLL 0x2090. Με αυτήν μόνο την πληροφορία ο αποκωδικοποιητής δεν μπορεί να επιλέξει σε ποια από τις δύο λέξεις αντιστοιχεί. Άρα δεν μπορεί να αποκωδικοποιηθεί στιγμιαία η ακολουθία μας. Βέβαια στην μια περίπτωση η ακολουθία έχει ως επόμενη κατάσταση την 2 ενώ στην άλλη την 3. Επειδή τα σύνολα των κωδικών λέξεων των καταστάσεων 2 και 3 είναι επιλεγμένα έτσι ώστε να διαφέρουν στα bit 15 και 3, μπορούμε κοιτάζοντας αυτά τα 2 bit της επόμενης κωδικής λέξης να αναγνωρίσουμε τη λέξη εισόδου.

Στο επόμενο κεφάλαιο θα παρουσιάσουμε την προσομοίωση της μεθόδου διαμόρφωσης και αποδιαμόρφωσης που υλοποιήσαμε η οποίες είναι βασισμένες στις μεθόδους της EFM plus.



Σχήμα 2.6: Αποδιαμόρφωση 8 προς 14 συν

Κεφάλαιο 3

Διαδικασία Διαμόρφωσης και Αποδιαμόρφωσης

Όπως αναφέραμε και στο προηγούμενο κεφάλαιο ο Imminck ανέπτυξε την μέθοδο EFM plus όπου χρησιμοποιούσε βιβλία κωδίκων τεσσάρων καταστάσεων για όλες τις 8 bit λέξεις εισόδου με βασικό και εναλλακτικό μέρος. Ακόμη μια τεχνική του ήταν και η εναλλαγή καταστάσεων (state switching) μεταξύ της κατάστασης 1 και 4. Στην υλοποίηση που θα παρουσιάσουμε στη συνέχεια έχουμε εξαλείψει την τεχνική του state switching πραγματοποιώντας αλλαγές στο βιβλίο κωδίκων.

3.1 Διαχωρισμός Καταστάσεων

Μέχρι τώρα τα βιβλία που χρησιμοποιήθηκαν ήταν τεσσάρων καταστάσεων με βασικό και εναλλακτικό μέρος για κάθε ακολουθία 8 bit. Οι καταστάσεις αυτές χωρίζονταν σύμφωνα με το πλήθος των αρχικά μηδενικά των λέξεων που παράγονται καθώς και το πλήθος των τελικών μηδενικών που οδηγούν σε αυτές. Για να μην έχουμε πλέον εναλλαγή των καταστάσεων 1 και 4 θα πρέπει να επεκτείνουμε το βιβλίο κωδίκων μας. Με την εναλλαγή των καταστάσεων έχουμε ουσιαστικά περισσότερες επιλογές για τον κωδικοποιητή, άρα θα πρέπει να παρέχουμε εμείς τις επιλογές αυτές χωρίς να χρειάζεται να ελεγχθούν οι εναλλακτικές καταστάσεις.

Αρχικά θα επεκτείνουμε το βιβλίο δημιουργώντας 2 επιπλέον εναλλακτικά μέρη των 256 θέσεων καθώς και με τέσσερις ακόμα καταστάσεις. Οι καταστάσεις αυτές

θα είναι παράγωγα των καταστάσεων 1 και 4, διότι μεταξύ αυτών των καταστάσεων επιτρέπεται η εναλλαγή. Η κατάσταση 1 χωρίζεται σε 2 καταστάσεις την 1_0 και την 1_1 στην κατάσταση 1_0 θα οδηγούν όσες λέξεις δεν έχουν τελικά μηδενικά και ήδη έχουν ως επόμενη κατάσταση την 1 ενώ στην 1_1 όσες έχουν 1 τελικό μηδενικό. Από την άλλη η κατάσταση 4 θα χωριστεί σε τέσσερις επιμέρους καταστάσεις τις 4_6 , 4_7 , 4_8 και 4_9 . Σε κάθε μία από τις παραγόμενες καταστάσεις οδηγούν κώδικες έχουν 6, 7, 8, και 9 τελικά μηδενικά αντίστοιχα και η επόμενη κατάστασή του ήταν η 4. Τώρα πρέπει να μελετήσουμε ποιοι κώδικες θα μπουν ως επιπλέον εναλλακτικοί σε κάθε κατάσταση. Πρώτα από όλα να πούμε ότι με τον διαχωρισμό των καταστάσεων 1 και 4 μεταφέρουμε σε κάθε επιμέρους κατάσταση στο βασικό και το πρώτο εναλλακτικό αντίγραφο των κωδίκων που προϋπήρχαν στις καταστάσεις αυτές. Έπειτα για κάθε 8-bit ακολουθία ελέγχουμε τις καταστάσεις 1 και 4 στο βασικό και εναλλακτικό μέρος του βιβλίου. Αν στην κατάσταση 1 έχουμε ακολουθίες με πλήθος αρχικών μηδενικών $n \in [2, 5]$ αυτές θα προστεθούν ως εναλλακτικές στις παράγωγες καταστάσεις της 4. Αυτό οφείλεται στο γεγονός ότι οι κωδικές λέξεις που οδηγούν στην κατάσταση 4 έχουν $m \in [6, 9]$ τελικά μηδενικά. Άρα στη κατάσταση 4_9 οι ακολουθίες θα έχουν δυο αρχικά μηδενικά στην 4_8 το πολύ τρία στην 4_7 μέχρι τέσσερα και στην 4_6 μέχρι και 5 μηδενικά. Έτσι δεν παραβιάζεται ο περιορισμός μήκους (2,10). Τώρα για τις ακολουθίες της κατάστασης 4, όσες δεν έχουν αρχικά μηδενικά θα προστεθούν ως εναλλακτικές στην κατάσταση 1_0 ενώ όσες έχουν ένα στην 1_1 .

Για να κατανοήσουμε καλύτερα την διαδικασία επέκτασης του βιβλίου κωδίκων ας δούμε το παρακάτω παράδειγμα.

Έστω ότι έχουμε τον Πίνακα 3.1 όπου βλέπουμε το βασικό και εναλλακτικό μέρος ενός βιβλίου κωδίκων.

Insert code	State 1	Next state	State 2	Next state	State 3	Next state	State 4	Next state
0	0x2009	1	0x4024	2	0x2009	1	0x4024	2
alt								
0	0x0480,	4	0x0480	4	0x4848	2	0x4848	2

Πίνακας 3.1: Βασικό και εναλλακτικό μέρος βιβλίου κωδίκων για την λέξη 0

Βήμα 1: Επέκταση του βιβλίου κωδίκων δημιουργώντας τις επιμέρους καταστάσεις των καταστάσεων 1 και 4.

Insert code	S.1 ₀	N S	S. 1 ₁	N S	S. 2	N S	S. 3	N S	S. 4 ₆	N S	S. 4 ₇	N S	S. 4 ₈	N S	S. 4 ₉	N S
0	0x2009	1	0x2009	1	0x4024	2	0x2009	1	0x4024	2	0x4024	2	0x4024	2	0x4024	2
0	0x0480	4	0x0480	4	0x0480	4	0x4848	2	0x4848	2	0x4848	2	0x4848	2	0x4848	2

Πίνακας 3.2: Αρχική επέκταση codebook

Βήμα 2: Αλλαγή των καταστάσεων μετάβασης σύμφωνα με τα καινούρια κριτήρια τελικών μηδενικών που αναφέραμε παραπάνω.

Insert code	S.1 ₀	N S	S. 1 ₁	N S	S. 2	N S	S. 3	N S	S. 4 ₆	N S	S. 4 ₇	N S	S. 4 ₈	N S	S. 4 ₉	N S
0	0x2009	1 ₀	0x2009	1 ₀	0x4024	2	0x2009	1 ₀	0x4024	2	0x4024	2	0x4024	2	0x4024	2
0	0x0480	4 ₇	0x0480	4 ₇	0x0480	4 ₇	0x4848	2	0x4848	2	0x4848	2	0x4848	2	0x4848	2

Πίνακας 3.3: Αλλαγή επόμενων καταστάσεων

Βήμα 3: Δημιουργία των επιπλέον εναλλακτικών βιβλίων ακολουθώντας τους κανόνες που ορίσαμε παραπάνω.

Insert code	S.1 ₀	N S	S. 1 ₁	N S	S. 2	N S	S. 3	N S	S. 4 ₆	N S	S. 4 ₇	N S	S. 4 ₈	N S	S. 4 ₉	N S
0	0x2009	1 ₀	0x2009	1 ₀	0x4024	2	0x2009	1 ₀	0x4024	2	0x4024	2	0x4024	2	0x4024	2
0	0x0480	4 ₇	0x0480	4 ₇	0x0480	4 ₇	0x4848	2	0x4848	2	0x4848	2	0x4848	2	0x4848	2
0	0x4024	2	0xFFFF	F	0xFFFF	F	0xFFFF	F	0x2009	1 ₀	0x2009	1 ₀	0x2009	1 ₀	0xFFFF	F
0	0x4848	2	0xFFFF	F	0xFFFF	F	0xFFFF	F	0x0480	4 ₇	0xFFFF	F	0xFFFF	F	0xFFFF	F

Πίνακας 3.4: Νέο βιβλίο κωδίκων

3.2 Νέο Κριτήριο Επιλογής Κωδικών Λέξεων

Το κριτήριο επιλογής κωδικής λέξης, που παρουσιάσαμε παραπάνω, προτάθηκε από τον Kees A. Schouhamer Immink και βρίσκει εφαρμογή στον κώδικα διαμόρφωσης EFM-plus [1]. Ουσιαστικά ο K.A.S. Immink επιλέγει την εκείνη κωδική λέξη που ελαχιστοποιεί κατ' απόλυτη τιμή την ποσότητα του RDS, έτσι όπως αυτή μετράται κάθε 16 bits, δηλαδή στο τέλος κάθε κωδικής λέξης.

Στον κώδικα διαμόρφωσης που εμείς σχεδιάσαμε χρησιμοποιήσαμε ένα αρκετά διαφορετικό κριτήριο. Η ιδέα για τη σχεδίαση ενός νέου κριτηρίου προήλθε μέσα από

την απλή παρατήρηση πως αφού κατά την ανάλυση Fourier για την εύρεση του φάσματος συχνοτήτων του κωδικοποιημένου σήματος λαμβάνονται υπόψη όλα τα NRZI ψηφία της ακολουθίας, ένα κριτήριο που λαμβάνει απόφαση εξετάζοντας τα ψηφία ανά 16 και αγνοώντας παντελώς τα ενδιάμεσα δεν μπορεί να είναι ορθό. Το νέο κριτήριο προτάθηκε από τον Ιωάννη Κατσαβουνίδη και αρχικά εφαρμόστηκε, σε συνεργασία με τη Μάρθα Πέππα, στο βιβλίο κωδικών του EFMplus δίνοντας πολύ καλά αποτελέσματα. Ουσιαστικά υπολογίζουμε το άθροισμα των τετραγώνων όλων των ενδιάμεσων RDS τιμών της κωδικής λέξης και επιλέγουμε αυτήν με το μικρότερο.

Έστω RDS' το τρέχον ψηφιακό άθροισμα και η κωδική λέξη η οποία θεωρώντας ότι αρχικά RDS = 0, δημιουργεί την RDS ακολουθία $X = X_{15}X_{14}...X_1X_0$ το άθροισμα των τετραγώνων στο τέλος της ακολουθίας θα είναι:

$$Sum_of_squares = (X_{15} + RDS')^2 + (X_{14} + RDS')^2 + \dots + (X_0 + RDS')^2 = (X_{15})^2 + (X_{14})^2 + \dots + (X_0)^2 + 2RDS'(X_{15} + X_{14} + \dots + X_0) + 16RDS'^2$$

Ωστόσο, για να κάνουμε καλύτερη πρόβλεψη της καλύτερης κωδικής λέξης δεν προσθέτουμε τον όρο $16RDS'^2$, αφού δεν επηρεάζει το αποτέλεσμα της σύγκρισης, αλλά υπολογίζουμε και χρησιμοποιούμε την νέα τιμή του RDS, έστω NEW_RDS, που προκύπτει από την κωδική λέξη X. Έτσι, η ποσότητα που χρησιμοποιεί το νέο κριτήριο είναι:

$$(X_{15})^2 + (X_{14})^2 + \dots + (X_0)^2 + 2RDS'(X_{15} + X_{14} + \dots + X_0) \quad (3.1)$$

Σε περίπτωση ισοπαλίας ο κωδικοποιητής επιλέγει τυχαία μία από της κωδικές λέξεις.

3.3 Μέθοδος Διαμόρφωσης

Με την έναρξη της κωδικοποίησης έχουμε μία 8-bit ακολουθία ως είσοδο στον κωδικοποιητή. Υπάρχει εδώ η σύμβαση ότι ο κωδικοποιητής βρίσκεται αρχικά στην κατάσταση I_0 και έχει ως προηγούμενο πρόσημο το θετικό, οπότε ελέγχει το βιβλίο κωδικών για την ακολουθία αυτή στο βασικό και εναλλακτικά μέρη του βιβλίου κωδικών. Εδώ θα πρέπει να υπενθυμίσουμε ότι με την κωδικοποίηση μίας ακολουθίας στο αρχείο γράφεται η NRZI αναπαράστασή της. Αφού συγκεντρωθούν οι υποψηφίες προς κωδικοποίηση RLL ακολουθίες, πρέπει να επιλεγεί η καταλληλότερη για εγγραφή στον δίσκο. Για να γίνει αυτό υπολογίζεται το τρέχον ψηφιακό άθροισμα τις κάθε NRZI αναπαράστασης

με τον τρόπο που έχουμε αναφέρει σε προηγούμενο κεφάλαιο. Ο έλεγχος μόνο των εναλλακτικών κάθε κατάστασης μας αρκεί, διότι έχουμε αφομοιώσει την εναλλαγή καταστάσεων χάρις στην επέκταση του βιβλίου κωδίκων.

Παράδειγμα :

Έστω ότι βρισκόμαστε στην κατάσταση I_0 με τελευταίο πρόσημο το συν και η λέξη εισόδου είναι το 0. Σύμφωνα με τον πίνακα 3.4 οι επιλογές του κωδικοποιητή είναι οι RLL **0x2009** και **0x0480**, αλλά όπως αναφέραμε και νωρίτερα μας ενδιαφέρουν μόνο οι NRZI αναπαραστάσεις των ακολουθιών αυτών. Άρα θα ασχοληθούμε με τις ακολουθίες **001111111110001** και **0000011100000000**. Για την επιλογή της καταλληλότερης προς εγγραφή ελέγχουμε τα τρέχον άθροισμα τετραγώνων στις ενδιάμεσες καταστάσεις που παρουσιάζεται παραπάνω (3.1). Στον Πίνακα 3.5 παρουσιάζεται ο υπολογισμός του αθροίσματος τετραγώνων για όλες τις ενδιάμεσες τιμές. Ως RDS έχουμε το άθροισμα $X_{15} + X_{14} + \dots + X_0$ και ως RDS^2 το $(X_{15})^2 + (X_{14})^2 + \dots + (X_0)^2$. Οπότε θα επιλέξουμε την ακολουθία 001111111110001 που παρέχει το μικρότερο Sum_of_squares θα θέσουμε ως τρέχον ψηφιακό άθροισμα το -6 το τελευταίο πρόσημο είναι πλέον το μείον (-) και θα θέσουμε την επόμενη κατάσταση δηλαδή την I_0 ως την επόμενη στην οποία θα γίνει ο έλεγχος για την επόμενη λέξη εισόδου.

RLL	NRZI	RDS	RDS ²	old RDS + newRDS	Sum_of_squares
0x2009	001111111110001	-56	356	0 +(-6)	1028
0x0480	0000011100000000	76	464	0 + 10	1984

Πίνακας 3.5: Υπολογισμός αθροίσματος τετραγώνων

Κατά τον Immink η επιλογή της καταλληλότερης NRZI γίνεται σύμφωνα με το ποια μας δίνει το μικρότερο κατ' απόλυτη τιμή RDS σε σχέση με την τρέχουσα τιμή του. Αναλυτικότερα στο παραπάνω παράδειγμα θα είχαμε πάλι την επιλογή της ίδιας ακολουθίας επειδή το τρέχον ψηφιακό άθροισμα κατ' απόλυτη τιμή είναι $|-6| = 6$ ενώ για την **0000011100000000** είναι το 10.

Algorithm 1 Διαδικασία Διαμόρφωσης

```
1:  $last\_sign \leftarrow 0$ 
2:  $current\_state \leftarrow 1_0$ 
3:  $RDS \leftarrow 0$ 
4: while There are codes do
5:   Read first insert_code
6:   for each basic and alternate NRZI for insert_code in current_state do
7:     Calculate Sum_of_squares from (3.1)
8:   end for
9:   Choose NRZI with minimum Sum_of_squares
10:   $RDS \leftarrow RDS + new\_RDS$ 
11:   $current\_state \leftarrow next\_state$ 
12:  Update last_sign
13: end while
```

3.4 Μέθοδος Αποδιαμόρφωσης

Στην αποδιαμόρφωση τόσο η εκτίμηση της επόμενης κατάστασης όσο και η αποκωδικοποίηση της τρέχουσας κωδικής λέξης γίνονται με μέγιστη πιθανοφάνεια, ενώ στην αποδιαμόρφωση του Immink γινόταν μόνο η αποκωδικοποίηση της τρέχουσας κωδικής λέξης.

Για την εκτίμηση της επόμενης κατάστασης δημιουργήσαμε ένα πίνακα με την ελάχιστη απόσταση Hamming για όλες τις πιθανές μεταβάσεις από μία κατάσταση σε μία άλλη για κάθε 17-bit ακολουθία που υπάρχει. Τώρα η αποκωδικοποίηση της τρέχουσας κωδικής λέξης γίνεται με την βοήθεια ενός ιστογράμματος που απεικονίζει, για τις 17-bit NRZI σε κάθε κατάσταση του βιβλίου κωδίκων, τις αποστάσεις Hamming με όλες τις πιθανές 17-bit ακολουθίες.

Παράδειγμα :

Έστω ότι έχουμε ως είσοδο στον αποκωδικοποιητή την 16-bit NRZI ακολουθία $+1+1-1-1-1-1-1-1+1+1+1-1-1-1-1$ την οποία για χάρη ευκολίας την αναπαριστούμε με 0 για τα +1 και 1 για τα -1, οπότε έχουμε την ακολουθία **0011111100011111**. Σύμφωνα με το βιβλίο κωδίκων του Πίνακα 2.1 η RLL ακολουθία από την οποία παράχθηκε η NRZI έχει απόσταση Hamming μηδέν με την NRZI αναπαράσταση της **0x2090** η οποία αντιστοιχεί στην θέση 4 και 8 στις καταστάσεις 1 και 2. Όμως ο αποκωδικοποιητής ελέγχει για όλες τις περιπτώσεις μετάβασης από μία κατάσταση i σε μία j τις αποστάσεις

Hamming με την ακολουθία μας, για την επεξηγήσει του παραδείγματός μας θα αναφερθούμε μόνο στις **0x2090** και **0x2120** με απόσταση 0 και 2. Για την **0x2090** παρατηρούμε ότι η μόνη διαφορά των δύο καταστάσεων είναι το next state τους δηλαδή η επόμενη κατάσταση. Στη θέση 4 έχουμε την 2 ως επόμενη κατάσταση ενώ στην 8 την 3. Από την άλλη πλευρά η **0x2120** έχει μόνο την 2 ως επόμενη κατάσταση. Για να γίνει σωστή εκτίμηση της κωδικής μας λέξης θα πρέπει να ελέγξουμε και την επόμενη ακολουθία του αρχείου μας. Έστω τώρα ότι η επόμενη ακολουθία είναι η **001111111000111** που αντιστοιχεί στην **0x2024** RLL. Εδώ θα πρέπει να βρούμε σε ποια από τις δύο επόμενες καταστάσεις αντιστοιχεί. Για την κατάσταση 2 έχουμε απόλυτη αντιστοίχιση σε δύο περιπτώσεις ενώ στην κατάσταση 3 η ακολουθίες με την μικρότερη απόσταση Hamming είναι η **111111111000111** και η **001111111000111** με απόσταση 2 δηλαδή οι **0x8024** και **0x2048** RLL. Άρα ο αποκωδικοποιητής έχει να διαλέξει μεταξύ έξι περιπτώσεων. Εάν επιλέξουμε την **0x2090** μπορεί να αποκωδικοποιηθεί στην λέξη 4 όταν βρισκόμαστε στις καταστάσεις 1 ή 2 με επόμενη κατάσταση την 2 και στην λέξη 8 με επόμενη κατάσταση την 3, ενώ στην περίπτωση που επιλέξουμε την **0x2120** θα έχει ως αποτέλεσμα την αποκωδικοποίησή της στην λέξη 5 με επόμενη κατάσταση την 2. Για να γίνει η σωστή επιλογή θα πρέπει να συγκρίνουμε το άθροισμα των ελάχιστων αποστάσεων Hamming των προς αποκωδικοποίηση ακολουθιών NRZI για όλες τις μεταβάσεις μεταξύ καταστάσεων και των αποστάσεων Hamming των επόμενων ακολουθιών στις επόμενες καταστάσεις που ορίζονται από τις τρέχουσες NRZI ακολουθίες. Δηλαδή :

$$Current_state_Hamming_distance_i + next_state_Hamming_distance_j \quad (3.2)$$

Στον Πίνακα 3.6 έχουμε τα αποτελέσματα του παραπάνω αθροίσματος για όλες τις μεταβάσεις. Βλέπουμε ότι οι συνδυασμοί με το ελάχιστο άθροισμα είναι εκείνοι που οδηγούν στη λέξη 4 την οποία και θα επιλέξει ο αποκωδικοποιητής.

codeword	state i ->state j	(3.2)	resulting code
0x2090	1 to 2	0 + 0	4
0x2090	1 to 3	0 + 2	8
0x2090	2 to 2	0 + 0	4
0x2090	2 to 3	0 + 2	8
0x2120	1 to 2	2 + 0	5
0x2120	2 to 2	2 + 0	5

Πίνακας 3.6: Υπολογισμός $Current_state_Hamming_distance_i + next_state_Hamming_distance_j$

Με την βοήθεια των πινάκων αυτών γίνεται η αποκωδικοποίηση μέγιστης πιθανοφάνειας εφικτή. Στον Αλγόριθμο 3.4 παρουσιάζουμε την διαδικασία αποδιαμόρφωσης

Algorithm 2 Διαδικασία Αποδιαμόρφωσης

```
1: Read first codeword
2: while There are codewords do
3:   Read next codeword
4:   for each current state i do
5:     for each next state j do
6:        $SUM = Hamming\_distance_{i,j} + Hamming\_distance_j$ 
7:     end for
8:   end for
9:   Choose minimum SUM
10:  Choose code to be demodulated
11:   $current\_codeword \leftarrow next\_codeword$ 
12: end while
```

3.5 Βελτίωση του Βιβλίου Κωδίκων του Standard DVD

Σε προηγούμενη μελέτη που έχει γίνει δημιουργήθηκαν βιβλία κωδίκων βελτιστοποιώντας αυτο που χρησιμοποιούσε ο Immink στο άρθρο του. Η βελτίωση έγινε βάση της μεταβλητότητας του RDS και οδήγησε σε αρκετά καλά αποτελέσματα [6]. Οι στρατηγικές που εφαρμόστηκαν ήταν ουσιαστικά εναλλαγές κωδικών λέξεων μεταξύ γραμμών και στηλών. Βέβαια η εναλλαγές αυτές δεν γίνονται αυθαίρετα, διέπονται από μία σειρά κανόνων που ορίζουν την εγκυρότητα ενός βιβλίου κωδίκων. Παρακάτω θα παρουσιάσουμε τους κανόνες αυτούς.

Όταν έχουμε μία κωδική λέξη σε μια κατάσταση X για να είναι συνεπές το βιβλίο κωδίκων μας θα πρέπει να υπάρχει για την ίδια λέξη εισόδου σε μία άλλη κατάσταση Y η ίδια κωδική λέξη.

Invalid codebook :

```
/*0*/0x1008, 3, 0x1024, 2, 0x8248, 3, 0x8248, 3,
/*1*/0x1024, 2, 0x1008, 3, 0x8904, 2, 0x8904, 2,
/*2*/0x2210, 2, 0x2210, 2, 0x9011, 1, 0x9011, 1,
```

Valid codebook :

**/*0*/0x1008, 3, 0x1008, 3, 0x8248, 3, 0x8248, 3,
/*1*/0x1024, 2, 0x1024, 2, 0x8904, 2, 0x8904, 2,
/*2*/0x2210, 2, 0x2210, 2, 0x9011, 1, 0x9011, 1,**

Τώρα όταν έχουμε δύο καταστάσεις που προήλθαν απο διαχωρισμό δεν γίνεται να υπάρχει η ίδια κωδική λέξη και στις δύο αυτές καταστάσεις.

Invalid codebook :

**/*0*/0x1008, 3, 0x1008, 3, 0x8248, 3, 0x8248, 3,
/*1*/0x1024, 2, 0x1024, 2, 0x1008, 2, 0x8904, 2,
/*2*/0x2210, 2, 0x2210, 2, 0x9011, 1, 0x9011, 1,**

Valid codebook :

**/*0*/ 0x1008, 3, 0x1008, 3, 0x8248, 3, 0x8248, 3,
/*1*/0x1024, 2, 0x1024, 2, 0x8904, 2, 0x8904, 2,
/*2*/0x1008, 3, 0x1008, 3, 0x9011, 1, 0x9011, 1,**

Τέλος θα πρέπει να υπάρχει για κάθε πιθανή κατάσταση για κάθε λέξη εισόδου κάποια έγκυρη μετάβαση. Σε περίπτωση που δεν υπάρχει το βιβλίο μας είναι άκυρο.

Invalid codebook :

**/*0*/ 0x1008, 3, 0x1008, 3, 0x8248, 3, 0x8248, 3,
/*1*/0x1024, 2, 0x1024, 2, 0x8904, 2, 0x8904, 2,
/*2*/0x2210, 2, 0xFFFF, 0, 0x9011, 1, 0x9011, 1,
.
.
.
/*0*/0x0489, 1, 0x4808, 2, 0x0489, 1, 0x4808, 2,
/*1*/0x0441, 1, 0x4890, 3, 0x0441, 1, 0x4890, 3,
/*2*/0x0480, 4, 0xFFFF, 0, 0x8111, 1, 0x8111, 1,**

Valid codebook :

```
/*0*/ 0x1008, 3, 0x1008, 3, 0x8248, 3, 0x8248, 3,  
/*1*/0x1024, 2, 0x1024, 2, 0x8904, 2, 0x8904, 2,  
/*2*/0x2210, 2, 0x2210, 2, 0x9011, 1, 0x9011, 1,  
.  
.  
.  
/*0*/0x0489, 1, 0x4808, 2, 0x0489, 1, 0x4808, 2,  
/*1*/0x0441, 1, 0x4890, 3, 0x0441, 1, 0x4890, 3,  
/*2*/0x0480, 4, 0xFFFF, 0, 0x8111, 1, 0x8111, 1,
```

Πέρα από τις βελτιστοποιήσεις που έγιναν στο βιβλίο κωδίκων του Standard DVD [6] για βελτιστοποίηση της μεταβλητότητας του RDS, μελετήσαμε τα βιβλία που παρουσίασαν καλύτερα αποτελέσματα στην ανθεκτικότητα στα σφάλματα από ότι αυτό του DVD. Προσπαθήσαμε να βρούμε κάποιο μοτίβο εναλλαγών που μπορεί να οδήγησε σε αυτήν την βελτίωση. Βέβαια η μορφολογία των παραγόμενων πιο ανθεκτικών βιβλίων είναι τέτοια που δυσκολεύει αρκετά την μελέτη των ενδιάμεσων εναλλαγών που οδήγησαν σε αυτά τα βιβλία. Έτσι βασιστήκαμε στη συνεισφορά της κάθε κωδικής λέξης στο υπολειπόμενο σφάλμα για την δημιουργία στρατηγικής εναλλαγών λέξεων.

Η εναλλαγή λέξεων που βρίσκονται στις καταστάσεις 1 και 4 δεν οδηγεί σε βελτιώσεις λόγω της ιδιότητας των καταστάσεων αυτών που επιτρέπει την εναλλαγή μεταξύ τους. Γι' αυτόν τον λόγο επικεντρωθήκαμε στις λέξεις που βρίσκονται στις καταστάσεις 2 και 3. Μετά την ανάλυση που κάναμε στο επόμενο κεφάλαιο για την ανθεκτικότητα σε σφάλματα, επιλέξαμε τις λέξεις με το μεγαλύτερο υπολειπόμενο σφάλμα στις καταστάσεις 2 και 3 και δοκιμάζοντας εναλλαγές με αρκετές επαναλήψεις καταλήξαμε σε μία περαιτέρω βελτίωση της τάξεως του 0,1 %.

Κεφάλαιο 4

Θεωρητική ανάλυση

Σε αυτό το κεφάλαιο θα αναφερθούμε σχετικά με την διαδικασία και τις μεθόδους που ακολουθήσαμε για τον υπολογισμό της διακύμανσης του RDS (variance of RDS) καθώς και για την ανθεκτικότητα σε σφάλματα (error resilience).

Όπως αναφέραμε και στο προηγούμενο κεφάλαιο, η μέχρι τώρα διαδικασία κωδικοποίησης- αποκωδικοποίησης γινόταν με βάση 16-bit κωδικές λέξεις. Εμείς εισάγαμε στην όλη διαδικασία την έννοια των 17 bit.

4.1 Θεωρητική Ανάλυση RDS

Κάθε βιβλίο κωδικών έχει μια συγκεκριμένη τιμή μεταβλητότητας RDS που μπορεί να μας δώσει. Αυτό έχει σχέση με την μορφολογία του βιβλίου. Για τον υπολογισμό αυτής θα πρέπει να μοντελοποιήσουμε την διαδικασία κωδικοποίησης ενός αρχείου.

4.1.1 Μοντελοποίηση Διαδικασίας Κωδικοποίησης

Σε αντίθεση με την διαδικασία κωδικοποίησης, όπου ένα αρχείο διαβάζετε ανά 8 bit και χρησιμοποιώντας ένα βιβλίο κωδικών μετατρέπεται σε μια 16-bit ακολουθία NRZI, εδώ θα πρέπει να προσομοιώσουμε όλες τις πιθανές 8-bit λέξεις εισόδου (256), για κάθε πιθανή μετάβαση από μία κατάσταση σε μία άλλη, για οποιαδήποτε μεταβολή τις τιμές του RDS και για κάθε μια από τις 2 περιπτώσεις του last sign (0 ή 1).

Έτσι έχουμε την δημιουργία των λεγόμενων Mega-states που αποτελούν το συνδυασμό όλων των παραπάνω μεταβάσεων. Ο αριθμός των Mega-states υπολογίζεται ως εξής:

$$\#Mega - state = \#lastsign * \#states * \#RDS_{,}values$$

Όπου last sign θα είναι 2, ο αριθμός των καταστάσεων είναι πλέον 8 και οι τιμές του RDS κυμαίνονται στο διάστημα [-40,+40] οπότε ο αριθμός τους είναι 41. Εδώ θα πρέπει να σημειώσουμε ότι ο αριθμός των διακυμάνσεων του RDS οφείλεται πρώτον στον ψαλιδισμό του RDS για να παραμένει στο διάστημα [-40,+40] και δεύτερον στην ιδιότητα των ακολουθιών RLL (2,10) που όταν παραχθούν οι αντίστοιχες NRZI το άθροισμα των ψηφίων τους είναι πάντα πολλαπλάσιο του 2.

Μέσω της διαδικασίας κωδικοποίησης υπολογίζουμε για κάθε μετάβαση από ένα Mega-state σε κάποιο άλλο την συνεισφορά του στη μεταβλητότητα του RDS(variance of RDS) καθώς και την πιθανότητα αυτής της μετάβασης. Οι δύο ποσότητες υπολογίζονται από τις παρακάτω εξισώσεις.

$$PMegastateX \rightarrow MegastateY = \sum_{n=0}^k \frac{n}{256}$$

$$Sum_of_squaresMegastateX \rightarrow MegastateY = \sum_{n=0}^k \frac{Sum_of_squares}{16*256}$$

Όπου k είναι ο αριθμός των μεταβάσεων από την κατάσταση X στην κατάσταση Y και Sum_of_squares είναι το άθροισμα των τετραγώνων όλων των ενδιάμεσων RDS της κωδικής λέξης.

Χρησιμοποιώντας τις παραπάνω έννοιες μπορούμε τώρα να υπολογίσουμε την θεωρητική τιμή του variance of RDS.

4.1.2 Στατική Κατανομή και Μεταβλητότητα RDS

Θεώρημα 1 (Μαρκοβιανή Αλυσίδα Διακριτού Χρόνου [9]). *Μια στοχαστική διαδικασία διακριτού χρόνου $X_{n \geq 0}$ λέγεται ότι είναι μια μαρκοβιανή αλυσίδα, αν ικανοποιούνται οι εξής δύο συνθήκες:*

(i) *Όλες οι τυχαίες μεταβλητές $X_{n \geq 0}$ παίρνουν τιμές στο ίδιο αριθμησιμο σύνολο καταστάσεων S .*

(ii) Για κάθε $n = 0, 1, \dots$, για κάθε $si, si0, si1, \dots, si(n-1), sj \in S$, ισχύει η παρακάτω ιδιότητα Μαρκόβ:

$$P(X_{n+1} = sj | X_n = si, X_{n-1} = sin-1, \dots, X1 = si(1), X0 = si(0)) = P(Xn + 1 = sj | Xn = si)$$

Αν έχουμε μια ομογενή Μαρκοβιανή Αλυσίδα, που η διαδικασία της περιγράφεται από έναν ενιαίο και ανεξάρτητο από το χρόνο πίνακα p_{ij} , τότε το διάνυσμα π ονομάζεται στατική κατανομή [9](η "αναλλοίωτο μέτρο") αν $\forall j \in S$ ισχύει ότι

$$0 \leq \pi_j \leq 1$$

$$\sum_{j \in S} \pi_j = 1$$

$$\pi_j = \sum_{i \in S} \pi_i p_{ij}$$

Όπου p_{ij} είναι η πιθανότητα μετάβασης από μία κατάσταση i σε μία j . Για τον υπολογισμό της στατικής κατανομής λύνουμε το παραπάνω σύστημα μέχρι το error να είναι μικρότερο του 10^{-16} ($\sum_{k=0}^{256} \pi(y) - \pi(x) < 10^{-16}$)

Έχοντας υπολογίσει τη μεταβλητότητα του RDS για κάθε μετάβαση, δημιουργήσαμε τον πίνακα $Mega_state_varianceRDS(i, j)$ όπου σε κάθε θέση του έχει το $Sum_of_squares(MegastateX \rightarrow MegastateY)$. Άρα για τον υπολογισμό της τελικής διασποράς του RDS χρησιμοποιούμε την σχέση :

$$Variance_of_RDS = \sum_{i=0}^{656} \pi_i * \sum_{j=0}^{656} Mega_state_VarianceRDS - ij$$

Στη συνέχεια θα αναλύσουμε το θεωρητικό υπόβαθρο που μας οδήγησε στην αξιολόγηση της ανθεκτικότητας σε σφάλματα κάθε βιβλίου κωδίκων.

4.2 Θεωρητική Ανάλυση Ανθεκτικότητας σε Σφάλματα

Σε αυτό το κεφάλαιο θα αναλύσουμε την διαδικασία με την οποία θα υπολογίζουμε το error resilience ενός βιβλίου κωδίκων. Όπως ήδη έχουμε αναφέρει το error resilience είναι ο λόγος των υπολειπόμενων εσφαλμένων λέξεων (8 bit) προς το πλήθος των αλλοιωμένων

κωδικών λέξεων(16 bit). Η διαδικασία μας χωρίζεται σε δύο στάδια. Στο πρώτο στάδιο προσδιορίζουμε το υπολειπόμενο σφάλμα από τις περιπτώσεις που προκύπτει αλλοιωμένη κωδική λέξη κατά ένα ή δύο bits και στο δεύτερο συνυπολογίζουμε και τις αλλοιώσεις που προήλθαν από περισσότερα από 2 εσφαλμένα bits.

4.2.1 Υπολογισμός Πιθανότητας Σφάλματος από Αλλοίωση ενός Bit

Όπως αναφέραμε και στο δεύτερο κεφάλαιο τα βιβλία κωδικών αποτελούνται από 16 bit ακολουθίες RLL που υπακούν στον περιορισμό μήκους $(d,k) = (2,10)$, όμως κατά την κωδικοποίηση μετατρέπονται στις αντίστοιχες NRZI πριν την εγγραφή τους σε κάποιο οπτικού δίσκο.

Κατά την αποκωδικοποίηση μιας αλλοιωμένης κωδικής λέξης ο αποκωδικοποιητής μπορεί να οδηγηθεί σε λανθασμένο αποτέλεσμα. Αυτό οφείλετε κυρίως στην ύπαρξη παραπάνω από μια υποψηφίων λέξεων με απόσταση Hamming μικρότερη του δύο. Έστω λοιπόν ότι έχουμε τις εξής ακολουθίες NRZI προς αποκωδικοποίηση **0x1FF0** , **0x0FF0**, **0x3FE0** όπου κάθε μία αντιστοιχεί σε μία διαφορετική 8 bit ακολουθία από το βιβλίο κωδικών. Αν υποθέσουμε ότι για να γίνει σωστά η αποκωδικοποίηση θα πρέπει μια επιλεχθεί η πρώτη ακολουθία στην διαδικασία επιλογής ο decoder θα έχει άλλες δύο υποψήφια που διαφέρουν η μια στο τέταρτο και η άλλη στο τρίτο most significant bit. Η ικανότητα του decoder να επιλέξει την «σωστή» λέξη μας δείχνει πόσο error resilient είναι το σύστημα μας.

Ο συνολικός αριθμός των 16 bit RLL ακολουθιών με περιορισμό μήκους $(2,10)$ είναι 566. Σε πολλές περιπτώσεις τα βιβλία κωδικών δεν χρησιμοποιούν όλες τις ακολουθίες. Για να υπολογίσουμε την πιθανότητα εμφάνισης της κάθε λέξης θα επιστρέψουμε λίγο στην διαδικασία της κωδικοποίησης. Κατά την διάρκεια αυτής της διαδικασίας ελέγχουμε για κάθε λέξη που επιλεγόταν από τον encoder σε ποιο Mega-state βρισκόταν. Οπότε με την χρήση της στατικής κατανομής της μπορούμε πλέον να υπολογίσουμε την πιθανότητα εμφάνισης κάθε κωδικής λέξης

$$P_{\text{codeword}} = \sum_{k=0}^{656} \frac{\text{Megastate_count}_k * p_k}{256}$$

ως Mega state-count ορίζουμε τον πλήθος των εμφανίσεων της λέξης σε κάθε Mega-state.

Με την ολοκλήρωση και αυτής της διαδικασίας σειρά έχει ο υπολογισμός του σφάλματος των code-word. Όπως αναφέραμε κα προηγουμένους στο σημείο αυτό ασχολού-

μαστε με τα σφάλματα που προέρχονται από λανθασμένη αποκωδικοποίηση εξαιτίας κάποιας διαφοράς bit μεταξύ των υποψήφιων λέξεων, συγκεκριμένα αναφερόμαστε στις διαφορές ενός ή δύο bit. Πιο αναλυτικά για κάθε 16 bit ακολουθία υπάρχουν 16 παράγωγες της αλλοιωμένες κατά 1 bit, όσες από αυτές συνεχίζουν να υπακούν στους κανόνες περιορισμού και υπάρχουν και στο βιβλίο κωδίκων με επαναφορά του αλλοιωμένου bit θα έχουμε πλήρη αντιστοίχιση με την «σωστή» κωδική λέξη. Ανάλογα αποτελέσματα θα έχουμε και στην περίπτωση που μία κωδική λέξη διαφέρει κατά δύο bit από μία άλλη, έτσι με την επαναφορά ενός εκ των δύο αλλοιωμένων bit θα προκύψει μια ακολουθία κατά ένα bit αλλοιωμένη από την «σωστή».

Αρχικά για να υπολογίσουμε το σφάλμα που προέρχεται από λέξεις με ένα bit αλλοιωμένο, συγκρίνουμε κάθε κωδική λέξη με όλες τις υπόλοιπες του βιβλίου και όσες έχουν hamming distance ένα ή δύο κρατάμε τις θέσεις των bit στις οποίες διαφέρουν από την αρχική μας ακολουθία. Αφού γίνει ολοκληρωμένος έλεγχος το υπολειπόμενο σφάλμα που οφείλεται στην αλλοίωση ενός bit υπολογίζεται ως εξής

$$P_{1_error} = \frac{n}{17}$$

με το **n** να αναπαριστά το πλήθος των θέσεων μεταξύ των 17 που είχαμε αλλοιωμένο bit για Hamming distance ένα..

Την πληροφορία που συλλέξαμε από τον προηγούμενο έλεγχο θα την χρησιμοποιήσουμε για τον υπολογισμό του σφάλματος εξαιτίας 2 αλλοιωμένων bit.

4.2.2 Υπολογισμός Πιθανότητας Σφάλματος από Αλλοίωση ενός ή δύο Bit

Για τον υπολογισμό του σφάλματος αυτού θα πρέπει να λάβουμε υπόψιν ότι μία λέξη που έχει μεγαλύτερη πιθανότητα εμφάνισης στο προς αποκωδικοποίηση bit-stream θα έχει πλεονέκτημα στην επιλογή της από τον decoder. Έχοντας αυτή την αρχή ως γνώμονα θα μπορούσαμε να εξηγήσουμε τον τρόπο αξιολόγησης του σφάλματος της κάθε λέξης.

Όπως αναφέραμε παραπάνω έχουμε συγκεντρώσει την πληροφορία που θα μας οδηγήσει στο υπολειπόμενο σφάλμα εξαιτίας της αλλοίωσης δύο ψηφίων, δηλαδή ποιες λέξεις διαφέρουν κατά δύο ψηφία και ποια είναι αυτά. Άρα για τον υπολογισμό του σφάλματος κάθε λέξης ελέγχουμε αρχικά την πιθανότητα της από κάθε άλλη με την οποία διαφέρει.

Σε περίπτωση που η πιθανότητα της "λάθος" λέξης είναι μεγαλύτερη τότε τα ψηφία που διαφέρουν μαρκάρονται και με την ολοκλήρωση του ελέγχου της κάθε λέξης με τις υπόλοιπες καταλήγουμε σε μία ποσότητα που αντιπροσωπεύει ένα μέρος του σφάλματος. Η ποσότητα αυτή είναι ίση με τον λόγο των μαρκαρισμένων ψηφίων προς τον συνολικό αριθμό ψηφίων κάθε λέξης που στην περίπτωσή μας είναι 17. Τώρα για όσες από αυτές έχουν την ίδια πιθανότητα η διαδικασία διαφέρει αρκετά. Κατά τον έλεγχο κάθε λέξης αντί τώρα να μαρκάρονται τα ψηφία, αθροίζουμε τον αριθμό τους για κάθε μία από τις 17 θέσεις bit που έχει κάθε κωδική λέξη. Σε αυτό το σημείο θα πρέπει να μην συνυπολογίσουμε τα ψηφία που προσδιορίσαμε στον υπολογισμό του σφάλματος σε περίπτωση μεγαλύτερης πιθανότητας της λανθασμένης κωδικής λέξης καθώς και εκείνα που προήλθαν από την αλλοίωση ενός ψηφίου που αναφερθήκαμε παραπάνω διότι θα οδηγηθούμε σε επικάλυψη των αποτελεσμάτων. Αυτό θα έχει ως αποτέλεσμα την ανακριβή προς τα πάνω αξιολόγηση του σφάλματος. Το συνολικό σφάλμα από την αλλοίωση ενός ή δύο ψηφίων προέρχεται από το άθροισμα των παραπάνω εννοιών και δίνεται από την εξής παράσταση

$$P_{2_error} = \frac{n + \sum_{equalprobabilities} \frac{n_2}{n_2+1}}{17}$$
$$P_{2_error} = P_{1_error} + \frac{\sum_{equalprobabilities} \frac{n_2}{n_2+1}}{17}$$

Όπου n είναι το πλήθος των μαρκαρισμένων bit σε περίπτωση αλλοίωσης ενός bit και n_2 είναι το άθροισμα των αλλοιωμένων στοιχείων για κάθε μία από τις θέσεις bit.

Με την ίδια διαδικασία υπολογισμού του σφάλματος ενός και δύο bit υπολογίζουμε και το σφάλμα από λάθος εκτίμηση της επόμενης κατάστασης μόνο που αντί να εφαρμόζουμε

την διαδικασία για όλες της κωδικές λέξης την εφαρμόζουμε μόνο σε αυτές των καταστάσεων 2 και 3.

$$P_{1,2_error} = \sum_{each\,codeword}^{max} (P_{1_error} + P_{2_error} + next_state_error) * P_{codeword}$$

Έχοντας υπολογίσει το υπολειπόμενο σφάλμα εξαιτίας ενός και δύο ψηφίων μπορούμε τώρα να προσδιορίσουμε το συνολικό υπολειπόμενο σφάλμα.

4.2.3 Συνολικό Υπολειπόμενο Σφάλμα

Στην διαδικασία που αναλύσαμε μέχρι τώρα δεν αναφερόμαστε πουθενά στην πιθανότητα αλλοίωσης κάποιου ψηφίου. Θα αναφερόμαστε σε αυτήν την τιμή ως P_{error} και θα κυμαίνεται από 10^{-4} έως και 10^{-2} . Το συνολικό υπολειπόμενο σφάλμα ορίζεται ως εξής:

$$residual\ error = \frac{\#faulty\ 16-bit\ demodulated\ codewords}{\#faulty\ 8-bit\ codewords\ to\ be\ demodulated}$$

Αρα η πιθανότητα να μην έχουμε λάθος σε κάποιο από τα 17 bits θα είναι :

$$P(0_error_in_17 - bits) = (1 - P_{error})^{17}$$

Με την ίδια λογική η πιθανότητα να έχουμε ένα λάθος θα είναι :

$$P(1_error_in_17 - bits) = (1 - P_{error})^{16} * P(error) * 17$$

Για 2 λάθη

$$P(2_error_in_17 - bits) = (1 - P_{error})^{15} * P(error) * 17$$

Αρα η πιθανότητα να έχουμε n λάθη σε 17 bit είναι:

$$P(n_error_in_17 - bits) = P((n - 1)_error_in_17 - bits) * \frac{P_{error}}{1 - P_{error}} * \frac{18 - n}{n}$$

Στην προηγούμενη ενότητα υπολογίσαμε στο υπολειπόμενο σφάλμα που προέρχεται από την αλλοίωση ενός ή και 2 bit. Θα αναφερόμαστε πλέον σε αυτήν την ποσότητα ως $Presidual_{1,2_error}$. Άρα η πιθανότητα να μην έχουμε ένα λανθασμένο bit σε μία 17-bit ακολουθία μετά την αποκωδικοποίηση είναι $1 - Presidual_{1,2_error}$ για δύο λανθασμένα bit είναι $1 - Presidual_{1,2_error}^2$ οπότε για n λανθασμένα bit η πιθανότητα θα είναι $1 - Presidual_{1,2_error}^n$.

Έχοντας κατανοήσει αυτές τις ποσότητες μπορούμε να ορίσουμε το συνολικό υπολειπόμενο σφάλμα ως τον λόγο του αθροίσματος του γινομένου, της πιθανότητας να είναι κάποιο bit αλλοιωμένο πριν την αποκωδικοποίηση επί την πιθανότητα να μην είναι λανθασμένο μετά την αποκωδικοποίηση για κάθε bit, προς το άθροισμα της πιθανότητα να μην είναι κάποιο λανθασμένο μετά την αποκωδικοποίηση. Η παραπάνω έννοια δίνεται στην σχέση:

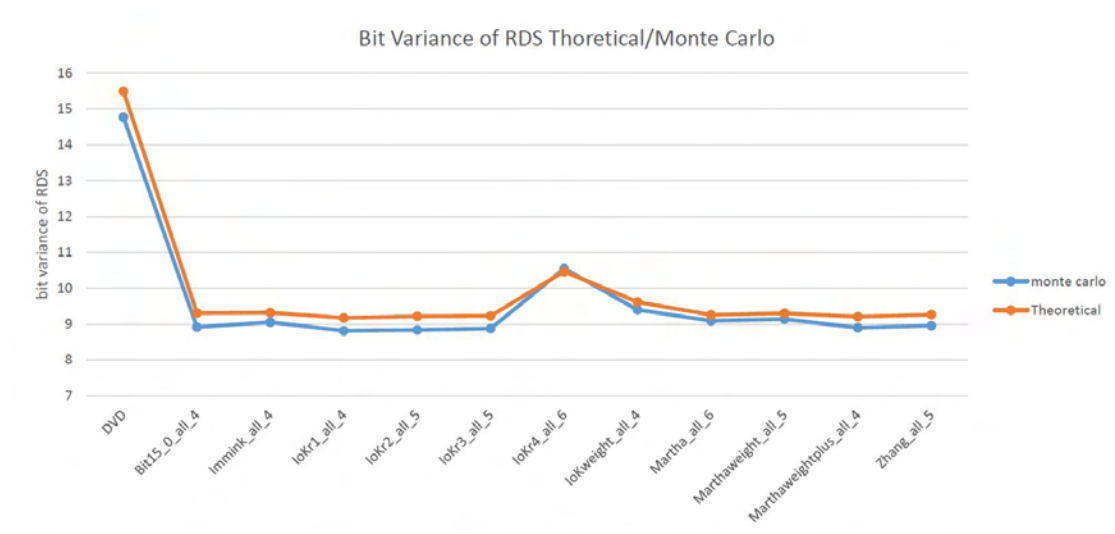
$$residual_error = \frac{\sum_{n=1}^{17} P(n_error_in_17-bits) * 1 - Presidual_{1,2_error}^n}{\sum_{n=1}^{17} 1 - Presidual_{1,2_error}^n}$$

Κεφάλαιο 5

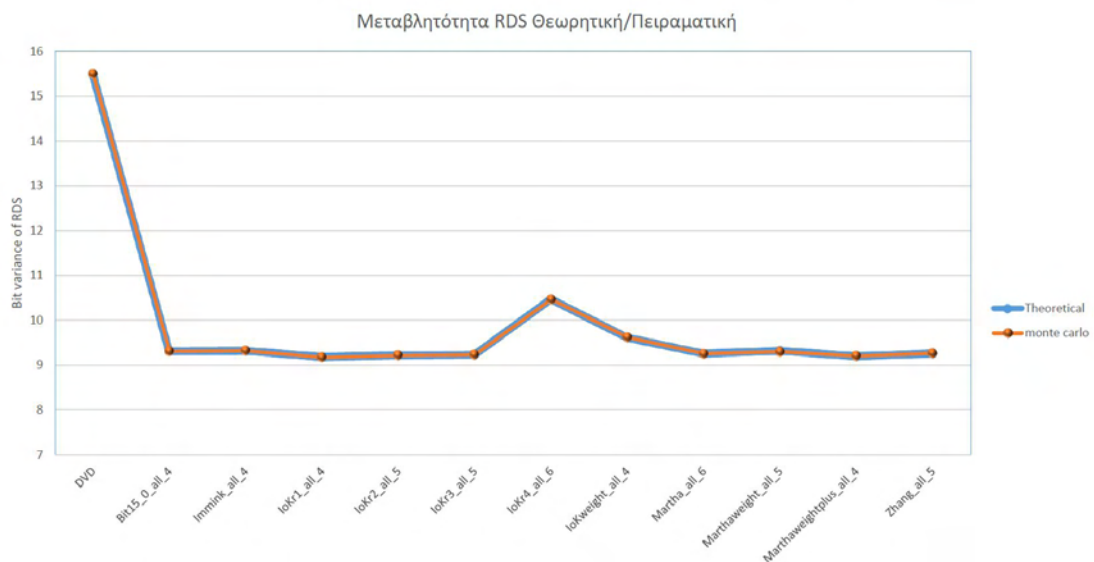
Αποτελέσματα

5.1 Αποτελέσματα Μεταβλητότητας RDS

Για τα πειραματικά αποτελέσματα χρησιμοποιήσαμε την τεχνική Monte Carlo[1] κατά την οποία εισάγουμε ένα τυχαίο αρχείο στην διαδικασία μας και κάνουμε μετρήσεις για να εξάγουμε την μεταβλητότητα του RDS και την ανθεκτικότητα στα σφάλματα(error resilience). Τα αρχεία που χρησιμοποιήσαμε ήταν δύο το Akina και Random των 3502 KB και 3072 KB αντίστοιχα. Στις δύο παρακάτω γραφικές παραστάσεις βλέπουμε την βελτίωση της μεταβλητότητας του RDS από το βιβλίο κωδίκων DVD που χρησιμοποιήθηκε στο άρθρο του Immink [1] έως και αυτό του Zhang που προήλθε από μία σειρά βελτιστοποιήσεων. Παρατηρούμε ότι για το αρχείο Random που αποτελεί ένα non-biased τυχαίο αρχείο τα αποτελέσματα της θεωρητικής ανάλυσης συμπίπτουν με αυτά των πειραματικών αποτελεσμάτων.



Σχήμα 5.1: Βελτίωση μεταβλητότητας RDS Θεωρητικής και πειραματικής με το αρχείο Akina

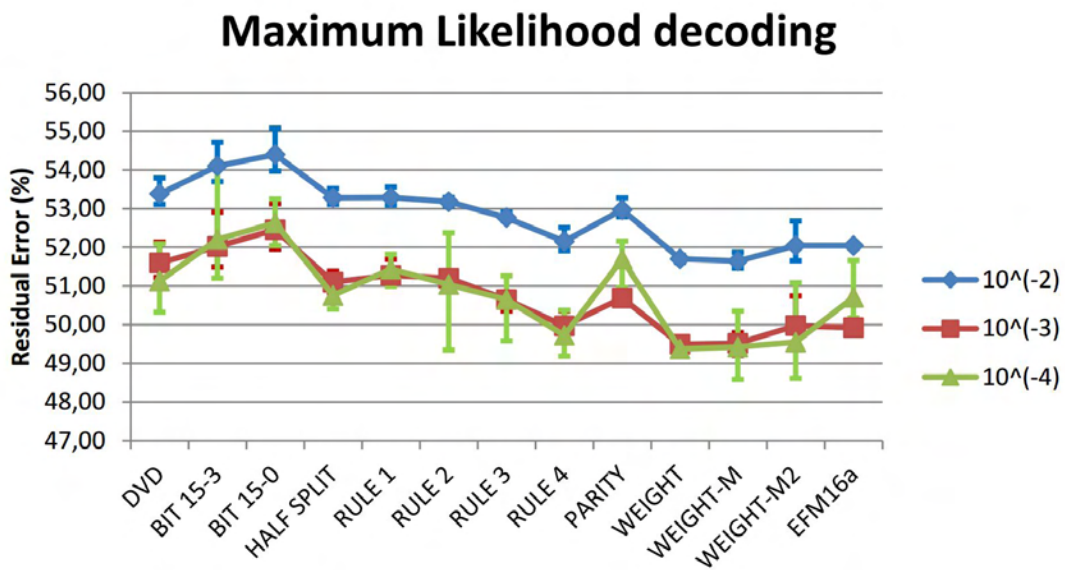


Σχήμα 5.2: Βελτίωση μεταβλητότητας RDS Θεωρητικής και πειραματικής με το αρχείο Random

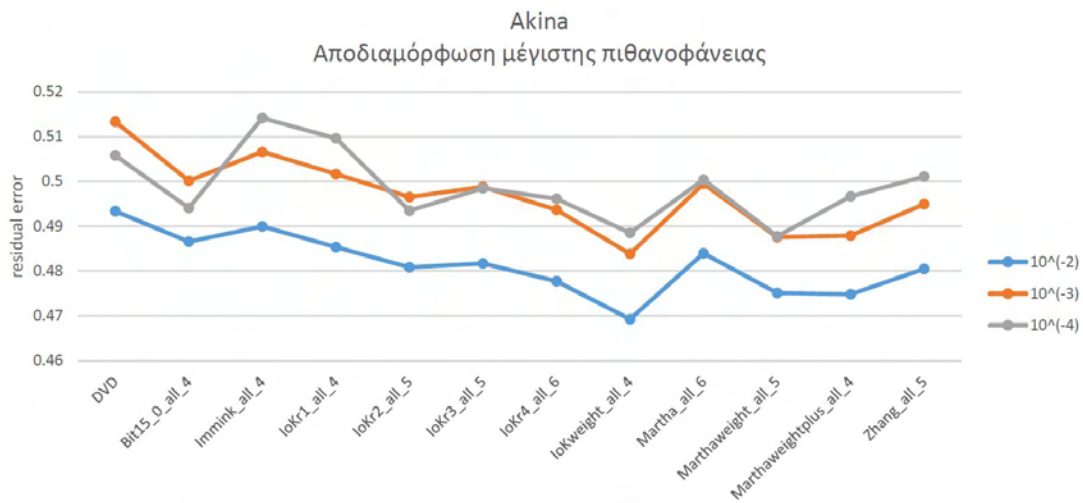
5.2 Αποτελέσματα Ανθεκτικότητας στα Σφάλματα

Στις παρακάτω γραφικές παραστάσεις δείχνουμε την ανθεκτικότητα στα σφάλματα του διαμορφωτή μετρώντας το ποσοστιαίο υπολειπόμενο σφάλμα. Έχουμε αλλοιώσει bit

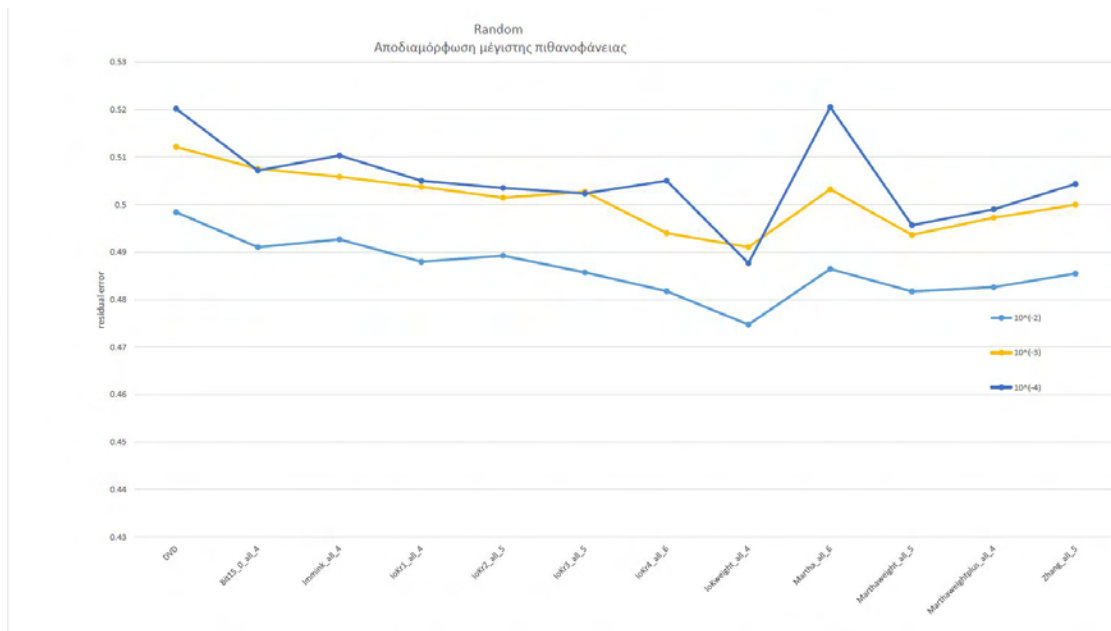
στο αρχείο με πιθανότητα 10^{-2} , 10^{-3} και 10^{-4} . Σε προηγούμενη μελέτη που είχε γίνει παρατηρήθηκε ότι με την αποκωδικοποίηση μέγιστης πιθανοφάνειας υπάρχουν τρόποι να οδηγηθούμε από το βιβλίο κωδίκων του Immink (DVD) σε άλλα πιο ανθεκτικά σε σφάλματα, αυτό φαίνεται και στο σχήμα 5.3. Τώρα χρησιμοποιώντας το πλεονέκτημα που μας δίνει η αποκωδικοποίηση μέγιστης πιθανοφάνειας μαζί με την χρησιμοποίηση 17 bit μπορούμε να πετύχουμε ακόμη μεγαλύτερη ανεκτικότητα. Τα αποτελέσματα στα δύο αρχεία που έχουμε φαίνονται στις γραφικές παραστάσεις στα σχήματα 5.5 και 5.4.



Σχήμα 5.3:

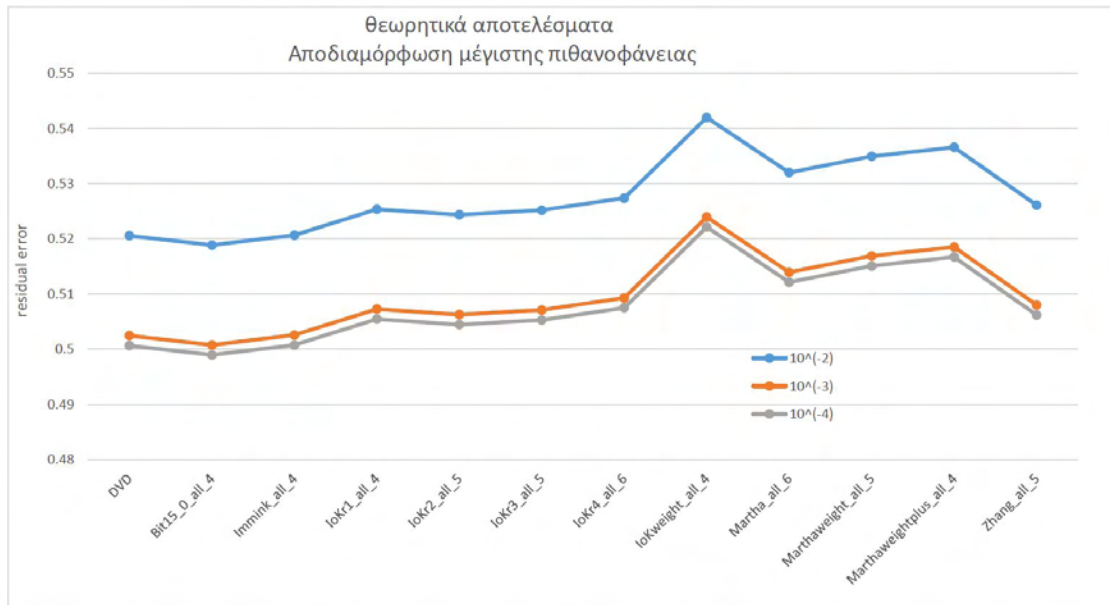


Σχήμα 5.4: Αποτελέσματα ανθεκτικότητας σε σφάλματα για το αρχείο Akina



Σχήμα 5.5: Αποτελέσματα ανθεκτικότητας σε σφάλματα για το αρχείο Random

Όλες οι διεργασίες που συνέβαλαν στην ολοκλήρωση της εργασίας μας εκτελέστηκαν σε τετραπύρνηνο μηχάνημα στα 2,4 GHz με 8 Gb RAM. Η συνολική διαδικασία κράτησε 9 δευτερόλεπτα με την θεωρητική ανάλυση να διαρκεί 0,5 δευτερόλεπτα. Στον παρακάτω πίνακα 5.2 βλέπουμε το ποσοστό χρήσης της κάθε διεργασίας.



Σχήμα 5.6: Θεωρητικά αποτελέσματα error resilience

Function Name	Functions work%
main	50.29
theoretical_analysis	6.33
find_all_cadeword_prob	0.24
calculate_codeword_error_resilience	0.01

Πίνακας 5.1: Ποσοστό χρήσης κάθε διεργασίας

Όπως είχαμε αναφέρει στην εισαγωγή στόχοι αυτής της εργασίας ήταν η μελέτη των μεθόδων διαμόρφωσης και αποδιαμόρφωσης με σκοπό να προσδιορίσουμε την μεταβλητότητα του RDS καθώς και την ανθεκτικότητα σε σφάλματα ενός βιβλίου κωδίκων. Η βελτιστοποίηση και περαιτέρω μελέτη της μεθόδου αποδιαμόρφωσης και τέλος η βελτιστοποίηση των βιβλίων κωδίκων για την παραγωγή νέων πιο ανθεκτικών.

Κεφάλαιο 6

Συμπεράσματα και Μελλοντική εργασία

Στην αρχή αυτής της εργασίας τέθηκαν οι στόχοι την δημιουργίας ενός εργαλείου που θα προσδιορίζει την ανθεκτικότητα σε σφάλματα και την μεταβλητότητα του RDS ενός βιβλίου κωδίκων και η προσπάθεια βελτίωσης του ως προς την ανθεκτικότητά του. Παρατηρώντας τα αποτελέσματα του προηγούμενου κεφαλαίου είδαμε ότι καταφέραμε η ανθεκτικότητα σε σφάλματα των βιβλίων κωδίκων βελτιώνεται με την βοήθεια του νέου τρόπου αποκωδικοποίησης που εισάγαμε. Η προσπάθεια μας να βελτιώσουμε ένα βιβλίο κωδίκων ανεξαρτήτως της μεθόδου αποκωδικοποίησης επέφερε κάποια θετικά αποτελέσματα αλλά χρειάζεται παραπάνω μελέτη για την ικανοποιητική βελτίωση τις ανθεκτικότητας τους.

Η έρευνα λοιπόν σε αυτόν τον τομέα δεν έχει ακόμα τελειώσει. Υπάρχουν στοιχεία που θα μπορούσαν να οδηγήσουν σε ακόμη καλύτερα αποτελέσματα και σε μεγαλύτερη βελτίωση των βιβλίων κωδίκων. Αρχικά θα μπορούσαμε να μελετήσουμε παραπάνω μεθόδους αποδιαμόρφωσης εκτός της αποδιαμόρφωσης μέγιστης πιθανοφάνειας, όπως η αποδιαμόρφωση Viterbi (Viterbi decoding) [8]. Ακόμη η μετέπειτα μελέτη των βιβλίων κωδίκων για επιπλέον στρατηγικές εναλλαγών κωδικών λέξεων μπορεί να οδηγήσει σε ακόμα μεγαλύτερη βελτιστοποίηση του error resilience.

Βιβλιογραφία

- [1] K.A.S Immink, “A Survey of Codes for Optical Disk Recording”, IEEE Journal Selected Areas in Communications, VOL. 19, pp. 756-764, April 2001
- [2] B.H. Marcus, P.H. Siegel, J.K. Wolf, “Finite-State Modulation Codes for Data Storage”, IEEE Journal Selected Areas in Communications, VOL 10, pp. 5-37, Jan. 1992
- [3] K.A.S Immink, “EFMPlus: The Coding Format of the Multimedia Compact Disc”, IEEE Transactions on Consumer Electronics, VOL. 41, pp. 491-497, August 1995
- [4] C. Shannon, “A Mathematical Theory of Communication”, Bell System Technical Journal, VOL 27, pp 379-423, 623-656, October 1948
- [5] R. Adler, D. Coppersmith, and M. Hassner, “Algorithms for sliding-block codes”, IEEE Transactions on Information Theory, VOL. IT-29, NO. 1, pp. 5-22, January 1983
- [6] Μ. Πέππα, Λ. Τασιούλας, Ι. Κατσαβουνίδης, “Μελέτη και υλοποίηση μεθόδων διαμόρφωσης καναλιού με περιορισμούς μήκους”, Διπλωματική Εργασία, Βόλος 2012
- [7] Χ. Αλιμήσης, Ι. Κατσαβουνίδης, Λ. Τασιούλας, “Βέλτιστη Διαμόρφωση Καναλιού με Περιορισμούς Μήκους και Χρήση Πολλαπλών Καταστάσεων”, Διπλωματική Εργασία Βόλος 2013
- [8] A.J. Viterbi, “Convolutional Codes and Their Performance in Communication Systems”, IEEE Transactions on Communications Technology, VOL. 19, pp. 751-771, October 1971
- [9] Ι.Τσιτσικλής, «Εισαγωγή στις πιθανότητες»
- [10] T, R.J., ”Analog to digital to optical photographic recording and playback system”, US Patent 3,501,586 , March 1970

- [11] Suzuki, T. and Sako, Y. and Furukawa, S. and Furuya, T. and Baggen, C.P.M.J., "Error correction system with cross-interleaved reed-solomon code", CA Patent 1,255,771, June 1989
- [12] Immink, K.A. and Nijboer, J.G. and Ogawa, H. and Odaka, K., "Method of coding binary data", US Patent 4,501,000, February 1985