



Πανεπιστήμιο Θεσσαλίας
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

**“Βέλτιστη Διαμόρφωση Καναλιού με Περιορισμούς Μήκους και
Χρήση Πολλαπλών Καταστάσεων”**

**"Optimized Run Length Limited Modulation Coding using Multiple
Stages"**

Διπλωματική Εργασία

Χρήστος Αλιμήσης

Επιβλέποντες Καθηγητές: Δρ. Ιωάννης Κατσαβουνίδης
Δρ. Λέανδρος Τασιούλας

Βόλος, Σεπτέμβριος 2013

Διπλωματική Εργασία για την απόκτηση του Διπλώματος του Μηχανικού Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων του Πανεπιστημίου Θεσσαλίας, στα πλαίσια του Προγράμματος Προπτυχιακών Σπουδών του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας.

.....

Χρήστος Δ. Αλιμής

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων Πανεπιστημίου Θεσσαλίας

Copyright © Christos Alimisis, 2013

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Θεσσαλίας.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά όλα τα μέλη της οικογένειάς μου, που συνέβαλαν στο να είμαι αυτός που σήμερα είμαι. Χωρίς αυτούς αδιαμφισβήτητα δε θα τα είχα καταφέρει. Ξεχωριστές ευχαριστίες στους γονείς μου και στις αδερφές μου για την υποστήριξη, συμπαράσταση, αλλά και υπομονή που έδειξαν όλα αυτά τα χρόνια.

Νιώθω επίσης την ανάγκη να εκφράσω τις ευχαριστίες μου και την εκτίμηση μου σε όλο το προσωπικό του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας που σε γενικές γραμμές επιτελεί ένα πολύ αξιόλογο διδακτικό και ερευνητικό έργο.

Θα ήταν σημαντικότερη παράλειψη αν δεν έκανα ειδική αναφορά στον κύριο Γιάννη Κατσαβουνίδη, ο οποίος υπήρξε ο βασικός εμπνευστής και επιβλέπωντας της διπλωματικής μου εργασίας. Η ολοκλήρωση της συνεργασίας μου μαζί του με αφήνει να δυσκολεύομαι να ξεχωρίσω αν εργαζόμενος δίπλα του αποκόμισα περισσότερα ως μηχανικός ή ως άνθρωπος.

Οφείλω ακόμη να εκφράσω και τις ευχαριστίες μου στον κύριο Λέανδρο Τασιούλα που δέχθηκε να συμμετάσχει στην εξεταστική επιτροπή αυτής της εργασίας, καθώς και στους κύριους Αντώνη Αργυρίου και Γεράσιμο Ποταμιάνο για κάποιες άριστες συνεργασίες που είχαμε στο παρελθόν.

Πολλά από τα ερευνητικά αποτελέσματα που παρουσιάζονται σε αυτήν την εργασία συνδιαμορφώθηκαν μέσα από την κοινή εργασία μας με την απόφοιτη του τμήματος Μάρθα Πέππα. Θα ήθελα να την ευχαριστήσω θερμά για την εξαιρετική συνεργασία που είχαμε όλο το προηγούμενο διάστημα και ειδικά τις τελευταίες μέρες του Αυγούστου.

Μόνο ως άψογη θα μπορούσα να χαρακτηρίσω και τη συνεργασία μου με τους διδακτορικούς φοιτητές Στέργιο Πουλαράκη και Δημήτρη Ζαχαρή. Να ευχαριστήσω και τον απόφοιτο του τμήματος Πέτρο Καλό για την τεχνική βοήθεια που μου προσέφερε κατά το κρίσιμο διάστημα του Ιουλίου.

Κλείνοντας θέλω να ευχαριστήσω όλα τα παιδιά του εργαστηρίου για τις πολύ ενδιαφέρουσες συζητήσεις μας τα βράδια της Τετάρτης και να αφιερώσω τη διπλωματική μου εργασία στην οικογένεια μου και στους φίλους μου.

Σας ευχαριστώ όλους από καρδιάς,
Χρήστος Αλιμήσης

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	ΕΙΣΑΓΩΓΗ/ΑΝΤΙΚΕΙΜΕΝΟ ΕΡΓΑΣΙΑΣ.....	2
2	ΔΙΑΜΟΡΦΩΣΗ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΕΓΓΡΑΦΗ ΣΕ ΟΠΤΙΚΟ ΔΙΣΚΟ.....	2
3	ΑΚΟΛΟΥΘΙΕΣ ΜΕ ΠΕΡΙΟΡΙΣΜΟΥΣ ΜΗΚΟΥΣ (d, k).....	4
4	ΤΡΕΧΟΝ ΨΗΦΙΑΚΟ ΑΘΡΟΙΣΜΑ.....	5
5	ΡΥΘΜΟΣ ΚΩΔΙΚΟΠΟΙΗΣΗΣ & ΧΩΡΗΤΙΚΟΤΗΤΑ ΚΑΝΑΛΙΟΥ.....	6
6	ΚΑΤΑΣΚΕΥΗ ΔΙΑΜΟΡΦΩΤΗ ΠΕΠΕΡΑΣΜΕΝΩΝ ΚΑΤΑΣΤΑΣΕΩΝ.....	7
7	ΣΥΝΘΗΚΗ ΑΘΡΟΙΣΜΑΤΟΣ ΓΡΑΜΜΗΣ.....	8
8	ΕΥΡΕΣΗ ΠΡΟΣΕΓΓΙΣΤΙΚΩΝ ΙΔΙΟΔΙΑΝΥΣΜΑΤΩΝ.....	9
9	ΣΥΓΧΩΝΕΥΣΗ ΚΑΤΑΣΤΑΣΕΩΝ.....	11
10	ΔΙΑΧΩΡΙΣΜΟΣ ΚΑΤΑΣΤΑΣΕΩΝ	14
11	ΣΥΝΕΠΗΣ ΑΝΑΘΕΣΗ ΚΩΔΙΚΩΝ ΛΕΞΕΩΝ ΣΕ ΛΕΞΕΙΣ ΕΙΣΟΔΟΥ.....	16
12	ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΜΟΡΦΩΣΗΣ/ΚΩΔΙΚΟΠΟΙΗΣΗΣ.....	19
13	ΔΙΑΔΙΚΑΣΙΑ ΑΠΟΔΙΑΜΟΡΦΩΣΗΣ/ΑΠΟΚΩΔΙΚΟΠΟΙΗΣΗΣ.....	22
14	ΘΕΩΡΗΤΙΚΟΣ ΥΠΟΛΟΓΙΣΜΟΣ RDS ΔΙΑΣΠΟΡΑΣ.....	24
15	ΒΕΛΤΙΩΣΗ ΚΩΔΙΚΑ ΔΙΑΜΟΡΦΩΣΗΣ.....	27
16	ΕΠΙΛΟΓΟΣ/ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΡΓΑΣΙΑΣ.....	34
17	ΑΝΑΦΟΡΕΣ.....	34

Εισαγωγή/Αντικείμενο Εργασίας

Στην παρούσα διπλωματική εργασία μελετήσαμε εκτενώς τις μεθόδους διαμόρφωσης των δεδομένων πριν την εγγραφή τους σε οπτικούς δίσκους. Τις εν λόγω μεθόδους ονομάζουμε Κώδικες Διαμόρφωσης Καναλιού (*Modulation Codes*). Η μελέτη μας επικεντρώθηκε στους κώδικες διαμόρφωσης EFM (*Eight to Fourteen Modulation*) και EFMplus, που βρίσκουν εφαρμογή στο CD (*Compact Disk*) και στο DVD (*Digital Versatile Disc*) αντίστοιχα. Στα πλαίσια αυτής της εργασίας σχεδιάστηκε ένας νέος κώδικας διαμόρφωσης καναλιού, στα πρότυπα του EFMplus. Στο παρόν κείμενο παρουσιάζουμε τον δικό μας κώδικα διαμόρφωσης καναλιού, ενώ παράλληλα κάνουμε συνοπτικές αναφορές στους κώδικες EFM και EFMplus. Αναφερόμαστε εκτενώς στα ζητήματα υλοποίησης του δικού μας κώδικα και κλείνουμε αξιολογώντας και συγκρίνοντας την απόδοση του με τον αντίστοιχο κώδικα EFMplus.

Διαμόρφωση Δεδομένων για Εγγραφή σε Οπτικό Δίσκο

Οι πρώτοι οπτικοί δίσκοι εφευρέθηκαν το 1958. Η εμπορική τους διάδοση στο ευρύ κοινό έγινε περίπου 20 χρόνια αργότερα, στις αρχές δηλαδή της δεκαετίας του 1980. Σήμερα οι οπτικοί δίσκοι, με κύριους εκπρόσωπους τους το CD και το DVD, συνεχίζουν να αποτελούν ένα επιτυχημένο μέσο αποθήκευσης ψηφιακής πληροφορίας. Το CD αναπτύχθηκε το 1968 από τον James Russell, με σκοπό την αποθήκευση και αναπαραγωγή ήχου. Έχει χωρητικότητα αποθήκευσης περίπου 700MB, με τα CD-R και CD-ROM, να είναι μετεξελίξεις του, ίδιας χωρητικότητας. Τα CD-R και CD-ROM διαφοροποιούνται ως προς το CD, καθώς έχουν τη δυνατότητα αποθήκευσης ψηφιακών δεδομένων γενικού σκοπού και όχι αποκλειστικά ήχου. Το DVD, που για πολλούς θεωρείται ο διάδοχος του CD, δημιουργήθηκε το 1995 και έχει τη δυνατότητα αποθήκευσης δεδομένων περί τα 4GB. Στον παρακάτω πίνακα συνοψίζονται τα βασικά τεχνικά χαρακτηριστικά του CD και του DVD.

Characteristic parameters of CD and DVD

Parameter	CD	DVD
Wavelength, nm	780	650
Numerical aperture	0.45/0.5	0.60/0.65
Track pitch, μm	1.6	0.74
Channel bit length, nm	277	133
Shortest effect length, nm	831	399
Modulation code	EFM	EFM+
Physical bit density, Gbit/inch ²	0.4	2.8
Reference disc velocity 1x CLV, m/s	1.2	4.0
Substrate/cover thickness, mm	1.2	0.6
Spot size, μm	0.9	0.55
Capacity per data layer, GB	0.65	4.7
Transfer rate at 1x speed, Mbit/s	4.3	11

Η εγγραφή ψηφιακής πληροφορίας σε έναν οπτικό δίσκο βασίζεται στη δημιουργία οπών(*pits*) επάνω στην επιφάνεια του δίσκου. Κατά την ολοκλήρωση της εγγραφής της πληροφορίας έχουν δημιουργηθεί στην επιφάνεια του δίσκου περιοχές οπών(*pits*) και περιοχές απουσίας οπών ή αλλιώς, περιοχές επιπέδων(*lands*). Κατά αντίστοιχο τρόπο, ο μηχανισμός ανάγνωσης βασίζεται στον εντοπισμό οπής ή επιπέδου από την ακτίνα ανάγνωσης. Για χάρη της συζήτησης, στην παρούσα εργασία θα αναφερόμαστε στο σήμα που διαβάζεται από την ακτίνα ανάγνωσης ως σήμα ανάγνωσης.

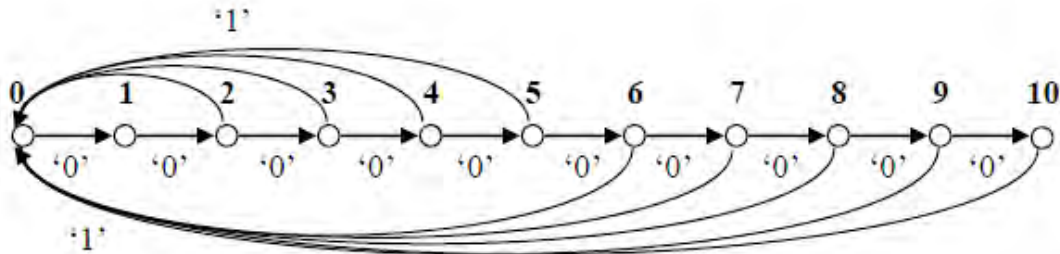
Η ύπαρξη πολύ μικρών ή πολύ μεγάλων οπών και επιπέδων στην επιφάνεια του δίσκου δυσκολεύει σημαντικά την αναπαραγωγή του. Ιδανικά θέλουμε το μέγεθος των οπών και των επιπέδων να μην είναι ούτε πολύ μεγάλο, αλλά ούτε και πολύ μικρό. Πολύ μεγάλο ή πολύ μικρό μέγεθος θα δυσκόλευε την ανάκτηση ρολογιού(*clock recovery*). Το ρολόι είναι αυτό που μετράει τις οπές και τα επίπεδα. Ένα λάθος στην εκτίμηση του πλήθους των ψηφίων που διαβάστηκαν μπορεί να οδηγήσει σε καταστροφικά αποτελέσματα, δηλαδή στην αδύνατη αναπαραγωγή του δίσκου. Για να λειτουργήσει σωστά το ρολόι, πρέπει να υπάρχουν συχνές εναλλαγές οπών και επιπέδων. Γι' αυτό το λόγο είναι επιθυμητό το μήκος των διαδοχικών οπών και το αντίστοιχο μήκος διαδοχικών επιπέδων να είναι εντός κάποιων ορίων. Επί προσθέτως πολύ μεγάλο μέγεθος οπής/επιπέδου θα οδηγούσε σε χαμηλής συχνότητας σήμα ανάγνωσης, το οποίο με τη σειρά του θα προκαλούσε παρεμβολή με το επίσης χαμηλής συχνότητας σήμα, που είναι υπεύθυνο για τον έλεγχο και τη διόρθωση της τροχιάς της ακτίνας ανάγνωσης.

Οι κώδικες διαμόρφωσης καναλιού είναι επιφορτισμένοι με το έργο της διαμόρφωσης των τυχαίων, δυαδικών ακολουθιών, ιδίου μήκους, που αντιπροσωπεύουν τα δεδομένα του χρήστη(έτσι όπως πιθανόν έχουν προκύψει από την κωδικοποίηση πηγής) σε νέες, συνήθως μεγαλύτερου μήκους δυαδικές ακολουθίες, οι οποίες υπακούουν σε ένα σύνολο περιορισμών. Θα αναφερόμαστε στις πρώτες ως δυαδικά ψηφία χρήστη(*user bits*) και στις δεύτερες ως δυαδικά ψηφία καναλιού(*channel bits*). Το επιθυμητό σύνολο των περιορισμών καθορίζεται αποκλειστικά από το σχεδιαζόμενο σύστημα επικοινωνίας ή σύστημα αποθήκευσης δεδομένων. Για την περίπτωση του συστήματος αποθήκευσης δεδομένων σε οπτικό δίσκο το σύνολο των περιορισμών θα μπορούσε να συνοψιστεί σε δύο μόνο ζητούμενα: (i) Ανάκτηση Ρολογιού και (ii) Σήμα Ανάγνωσης με Μικρή Ισχύ στις Χαμηλές Συχνότητες. Ονομάζουμε τις ακολουθίες με μικρή και ιδανικά μηδενική ισχύ στις χαμηλές συχνότητες *DC-free* ή *DC-balanced* ακολουθίες.



Ακολουθίες με Περιορισμούς Μήκους (d, k)

Οι κώδικες διαμόρφωσης καναλιού που μελετάμε βασίζονται στη χρήση RLL(*Run Length Limited*) ακολουθιών, που υπακούουν στον περιορισμό $(d, k) = (2, 10)$. Πρακτικά θέλουμε κάθε άσπος να ακολουθείται από τουλάχιστον 2 και μέχρι 10 μηδενικά. Για μεγαλύτερη κατανόηση δίνουμε παρακάτω το γράφο, που περιγράφει όλες τις έγκυρες (d, k) δυαδικές ακολουθίες.



Αν επιχειρήσουμε να «μαθηματικοποιήσουμε» τον παραπάνω γράφο, θα καταλήξουμε στον πίνακα γειτνίασης, που παρουσιάζεται παρακάτω. Ας ονομάσουμε τον πίνακα αυτόν A . Ο A θα περιλαμβάνει μόνο άσπους και μηδενικά και θα είναι διαστάσεων 11×11 , όσες είναι δηλαδή και οι καταστάσεις του παραπάνω γράφου.

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

Αριθμούμε τόσο τις γραμμές, όσο και τις στήλες του πίνακα με δείκτες που λαμβάνουν τιμές στο διακριτό διάστημα $[0-10]$. Με το στοιχείο $A[1][0]$ της δεύτερης γραμμής του πίνακα (γραμμή $A[1]$), που ισούται με μηδέν, να συμβολίζει την αδυναμία μετάβασης από την κατάσταση $[1]$ στην κατάσταση $[0]$. Ενώ το στοιχείο $A[1][2]$, που ισούται με 1, την ύπαρξη νόμιμης μετάβασης από την κατάσταση $[1]$ στην κατάσταση $[2]$.

Η ύψωση του πίνακα A σε μία δύναμη k , ας υποθέσουμε για χάρη συζήτησης πως $k=8$, θα μας δώσει τον παρακάτω πίνακα.

$$A^8 = \begin{pmatrix} 4 & 3 & 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 6 & 4 & 3 & 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 9 & 6 & 4 & 3 & 2 & 1 & 1 & 1 & 0 & 0 & 1 \\ 9 & 6 & 4 & 3 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 8 & 6 & 4 & 3 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 8 & 5 & 4 & 3 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 8 & 5 & 3 & 3 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 7 & 5 & 3 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ 6 & 4 & 3 & 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 5 & 3 & 2 & 2 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 3 & 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Το στοιχείο $A^8[1][0]$ της δεύτερης γραμμής του πίνακα(γραμμή $A^8[1]$), που ισούται πλέον με 6, συμβολίζει την ύπαρξη 6 διαφορετικών δυαδικών ακολουθιών μήκους 8, που υπακούουν στον περιορισμό $(d, k) = (2, 10)$, και μεταβαίνουν από την κατάσταση [1] στην κατάσταση [0].

Η χρήση RLL(*Run Length Limited*) ακολουθιών μας λύνει το σημαντικότερο πρόβλημα της ανάκτησης ρολογιού(*clock recovery*), ωστόσο δεν εγγυάται από μόνη της διαμόρφωση ακολουθιών μικρής ισχύος στις χαμηλές συχνότητες.

Τρέχον Ψηφιακό Άθροισμα

Στην εισαγωγή της εργασίας σημειώσαμε πως οι κώδικες διαμόρφωσης καναλιού αναλαμβάνουν να διαμορφώσουν τα *user bits* σε *channels bits*, με το πλήθος των δεύτερων να είναι συνήθως μεγαλύτερο των πρώτων. Για παράδειγμα, αν υποθέσουμε πως για τη δυαδική, μήκους 8, ακολουθία $[\text{bin}]11101101=[\text{dec}]237$, ο κώδικας καναλιού θα κατασκευάσει τη δυαδική, μήκους 16, ακολουθία $[\text{bin}]0000001001001001=[\text{dec}]585$. Η $[\text{bin}]0000001001001001=[\text{dec}]585$ είναι μια RLL ακολουθία που υπακούει στον περιορισμό $(d, k)=(2, 10)$. Πριν την εγγραφή της δυαδικής RLL ακολουθίας στον οπτικό δίσκο, η ακολουθία θα μετατραπεί σε μορφή NRZI(*non-return-to-zero*). Η NRZI ακολουθία αποτελείται μόνο από θετικούς και αρνητικούς άσσους. Κάθε RLL ακολουθία έχει δύο δυνατές απεικονίσεις σε NRZI μορφή. Τη θετική απεικόνιση, όπου ξεκινάμε με θετικό πρόσημο και αλλάζουμε την πολικότητα κάθε φορά που συναντάμε έναν άσσο στην RLL ακολουθία και την αντίστοιχη αρνητική, όπου εφαρμόζουμε την ίδια διαδικασία έχοντας όμως ξεκινήσει με αρνητικό πρόσημο. Για το παράδειγμά μας θα έχουμε:

Ακολουθία RLL: 0000001001001001

Ακολουθία NRZI[+]: +1+1+1+1+1+1-1-1+1+1+1-1-1+1

Ακολουθία NRZI[-]: -1-1-1-1-1+1+1+1-1-1+1+1+1-1

Ονομάζουμε το άθροισμα των ψηφίων της NRZI ακολουθίας σε μία συγκεκριμένη θέση Τρέχον Ψηφιακό Άθροισμα(*Running Digital Sum*). Αν θεωρήσουμε πως η θέση [0] σημειώνει το αριστερότερο ψηφίο της NRZI ακολουθίας, τότε μπορούμε να πούμε πως το RDS της θετικής NRZI ακολουθίας στη θέση [3] ισούται με

$RDS[+] = +1+1+1=3$, με το $RDS[-]$ στην ίδια θέση να ισούται με $RDS[-] = -1-1-1=-3$. Ισχύει δηλαδή πως $RDS[-] = -RDS[+]$. Συνήθως ενδιαφερόμαστε για την τιμή του RDS στο τέλος μιας ακολουθίας, δηλαδή στη θέση [15], όπου για το παράδειγμά μας έχουμε $RDS[+]=4$ και $RDS[-]=-4$.

Βασιζόμενοι σε όσα αναφέραμε παραπάνω, μπορούμε να φανταστούμε το σήμα ανάγνωσης ως μια πολύ μεγάλου μήκους NRZI ακολουθία, το πλήθος των ψηφίων της οποίας μας είναι γνωστό. Εάν τώρα θεωρήσουμε την ποσότητα του τρέχοντος ψηφιακού αθροίσματος (RDS) ως μία απλή τυχαία μεταβλητή, μπορούμε εύκολα να υπολογίσουμε την μέση τιμή της ($mean$) και την διασπορά της ($variance$).

Μας ενδιαφέρει η Διασπορά του Τρέχοντος Ψηφιακού Αθροίσματος ($RDS Variance$) να λαμβάνει όσο το δυνατόν μικρότερες τιμές. Πολλοί ερευνητές (αναφορές στα [3], [10], [11], [16] και [17]) έχουν δείξει πως μικρή τιμή της εν λόγω ποσότητας μας εξασφαλίζει σήμα ανάγνωσης με μικρή ισχύ στις χαμηλές συχνότητες. Για τους κώδικες διαμόρφωσης που μελετάμε, EFM και EFMplus, η διασπορά είναι 16 και 19 αντίστοιχα (αναφορές στο [17]).

Τούδε και στο εξής κάθε φορά που θα αναφερόμαστε στη Διασπορά του Τρέχοντος Ψηφιακού Αθροίσματος ($Variance of Running Digital Sum$) θα χρησιμοποιούμε τον όρο RDS Διασπορά.

Ρυθμός Κωδικοποίησης & Χωρητικότητα Καναλιού

Εάν θεωρήσουμε πως το πλήθος των $user bits$ ισούται με p και το αντίστοιχο των $channel bits$ με q , τότε μπορούμε να υπολογίσουμε το λόγο $p/q (=R)$, καθώς και την ποσότητα $1-R$. Ονομάζουμε το λόγο R ρυθμό κωδικοποίησης ($Coding Rate$) και την ποσότητα $1-R$ πλεονάζουσα πληροφορία του κώδικα ($Redundancy of the Code$).

Ο κώδικας διαμόρφωσης EFM ($Eight to Fourteen Modulation$) επιτυγχάνει ρυθμό κωδικοποίησης $8/17=0.47$, με τον EFMplus να λειτουργεί βελτιωτικά σε σχέση με τον προκάτοχο του επιτυγχάνοντας ρυθμό $8/16=0.5$. Ο ρυθμός κωδικοποίησης φράσσεται άνωθεν από τη χωρητικότητα καναλιού ή αλλιώς χωρητικότητα Shannon. Δηλαδή αν εκφράσουμε μαθηματικά το παραπάνω θα έχουμε:

$$p/q (=R) \leq \text{Channel_Capacity}$$

Ως βέλτιστος κώδικας ορίζεται θεωρητικά ο κώδικας εκείνος που επιτυγχάνει ρυθμό κωδικοποίησης ίσο με τη χωρητικότητα, ωστόσο η δουλειά των Ashley και Siegel στο [9] μας δείχνει πως είναι πρακτικά αδύνατη η κατασκευή ενός τέτοιου κώδικα. Ως στόχος λοιπόν τίθεται η κατασκευή ενός κώδικα, ο οποίος προσεγγίζει όσο το δυνατόν καλύτερα τη χωρητικότητα καναλιού.

Ο Shannon στο [5] έδειξε πως η χωρητικότητα καναλιού για κωδικοποιητές με περιορισμούς μήκους (d, k) μπορεί να υπολογιστεί εύκολα, εάν λογαριθμήσουμε τη μέγιστη ιδιοτιμή του πίνακα γειτνίασης του (d, k) κωδικοποιητή. Οπότε έχουμε:

$$C(d, k) = \log_2(\lambda_{dk})$$

Η ύπαρξη μίας τουλάχιστον θετικής ιδιοτιμής και ενός αντίστοιχου θετικά ορισμένου ιδιοδιανύσματος είναι εγγυημένη από τη θεωρία των Perron-Frobenius, της οποίας μια αναλυτική περιγραφή μπορεί να βρεθεί στο [15].

Η μέγιστη ιδιοτιμή του πίνακα γειτνίασης, έτσι όπως αυτός προκύπτει για ισχύ του περιορισμού $(d, k)=(2, 10)$, ισούται με $\lambda_{dk} = 1.4558$. Η χωρητικότητα καναλιού υπολογίζεται στα $C = \log_2(1.4558) = 0.5418$ bits.

Στον παρακάτω πίνακα δείχνουμε πως αλλάζει η χωρητικότητα για μεταβολή των παραμέτρων περιορισμού (d, k) .

$k \backslash d$	0	1	2	3	4	5	6
1	.6942						
2	.8791	.4057					
3	.9468	.5515	.2878				
4	.9752	.6174	.4057	.2232			
5	.9881	.6509	.4650	.3218	.1823		
6	.9942	.6690	.4979	.3746	.2669	.1542	
7	.9971	.6793	.5174	.4057	.3142	.2281	.1335
8	.9986	.6853	.5293	.4251	.3432	.2709	.1993
9	.9993	.6888	.5369	.4376	.3630	.2979	.2382
10	.9996	.6909	.5418	.4460	.3746	.3158	.2633
11	.9998	.6922	.5450	.4516	.3833	.3282	.2804
12	.9999	.6930	.5471	.4555	.3894	.3369	.2924
13	.9999	.6935	.5485	.4583	.3937	.3432	.3011
14	.9999	.6938	.5495	.4602	.3968	.3478	.3074
15	.9999	.6939	.5501	.4615	.3991	.3513	.3122
∞	1.0000	.6942	.5515	.4650	.4057	.3620	.3282

Έχοντας υπολογίσει τις ποσότητες R (Ρυθμός Κωδικοποίησης) και C (Χωρητικότητα Καναλιού) μπορούμε να υπολογίσουμε την ποσότητα $n=R/C$. Η ποσότητα n χρησιμοποιείται ως ένα κριτήριο μέτρησης της απόδοσης του κώδικα και ουσιαστικά μας δείχνει πόσο κοντά είναι ο ρυθμός κωδικοποίησης στη χωρητικότητα καναλιού.

Κατασκευή Διαμορφωτή Πεπερασμένων Καταστάσεων

Εάν θεωρήσουμε σύστημα περιορισμού S , με χωρητικότητα καναλιού (χωρητικότητα Shannon) $\text{Channel_Capacity}(S)$ και ρυθμό κωδικοποίησης $R=p/q$, με τους p και q να είναι θετικοί ακέραιοι, τότε εφόσον ικανοποιείται η ανισότητα

$$p/q(=R) \leq \text{Channel_Capacity}$$

είναι δυνατή η κατασκευή κωδικοποιητή πεπερασμένων καταστάσεων με αποκωδικοποιητή εξαρτώμενο από καταστάσεις, που υπακούει στον περιορισμό που ορίζει το σύστημα S και έχει ρυθμό κωδικοποίησης $R=p/q$.

Αναλυτικά η διατύπωση και απόδειξη του παραπάνω θεωρήματος μπορεί να βρεθεί στα [2] και [6].

Συνθήκη Αθροίσματος Γραμμής

Σημειώσαμε ήδη πως ο ρυθμός κωδικοποίησης ($R=p/q$) που επιτυγχάνεται από τον κώδικα διαμόρφωσης EFMplus είναι $8/16$ (άρα $p=8$ και $q=16$). Ο κώδικας που εμείς σχεδιάσαμε επιτυγχάνει ακριβώς τον ίδιο ρυθμό (rate). Παρακάτω παρουσιάζουμε τον πίνακα γειτνίασης A υψωμένο στη δύναμη $q=16$.

$A^{16} =$	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

Κατ' αντιστοιχία με όσα αναφέραμε στην εισαγωγή της εργασίας, στον πίνακα A^{16} , το στοιχείο $A^{16}[1][0]$ ισούται με 83 και συμβολίζει την ύπαρξη 83 διαφορετικών δυαδικών ακολουθιών μήκους 16, που υπακούουν στον περιορισμό $(d, k) = (2, 10)$, και μεταβαίνουν από την κατάσταση [1] στην κατάσταση [0]. Αν τώρα αθροίσουμε τα στοιχεία του πίνακα κατά γραμμές θα πάρουμε το παρακάτω διάνυσμα.

$$r = [262, 382, 556, 547, 534, 515, 487, 447, 389, 304, 180]$$

Κάθε ένα από τα στοιχεία του διανύσματος r αντιπροσωπεύει το πλήθος των κωδικών λέξεων που έχει η αντίστοιχη κατάσταση. Παραδείγματος χάριν το δεύτερο στοιχείο του διανύσματος ισούται με 382 και σηματοδοτεί την ύπαρξη 382 διαφορετικών κωδικών λέξεων στην κατάσταση [1]. Εφόσον βασίσουμε τη σχεδίαση του διαμορφωτή μας σε αυτόν τον πίνακα, σημαίνει πως η κατάσταση [1] θα διαθέτει 122 διαφορετικές κωδικές λέξεις που οδηγούν στην κατάσταση [0], 83 διαφορετικές κωδικές λέξεις που οδηγούν στην κατάσταση [1], 57 στην κατάσταση [2] και ούτω καθ' εξής.

Το επιτρεπτό πλήθος των κωδικών λέξεων κάθε κατάστασης φράσσεται κάτωθεν από το ρυθμός κωδικοποίησης R και πιο συγκεκριμένα από το πλήθος p των user bits. Όπως ήδη αναφέραμε ο κώδικας διαμόρφωσης διαμορφώνει $p (=8)$ user bits σε

$q(=16)$ channel bits, άρα κάθε κατάσταση οφείλει αναγκαστικά να έχει διαθέσιμες τουλάχιστον $2^8(=256)$ κωδικές λέξεις. Η παραπάνω απαίτηση για το πλήθος των κωδικών λέξεων κάθε κατάστασης είναι γνωστή στη βιβλιογραφία με τον όρο “*Συνθήκη Αθροίσματος Γραμμής*”.

Συνθήκη Αθροίσματος Γραμμής: Το άθροισμα των στοιχείων κάθε γραμμής του πίνακα γειτνίασης A , υψωμένου στη δύναμη q , πρέπει να είναι **μεγαλύτερο ή ίσο με 2^p** . Όπου p και q , ο αριθμός των *user* και των *channel bits* αντίστοιχα.

Παρατηρούμε πως όλα τα στοιχεία του διανύσματος r είναι μεγαλύτερα ή ίσα του 256. Εξάιρεση αποτελεί το τελευταίο στοιχείο, το οποίο δεν ικανοποιεί τη συνθήκη, αφού $r[10]=180 < 256$. Η μη ικανοποίηση της συνθήκης μας αναγκάζει είτε να προχωρήσουμε στη οριστική διαγραφή της κατάστασης [10], κατασκευάζοντας αντί για έναν διαμορφωτή 11 καταστάσεων, έναν αντίστοιχο με 10 καταστάσεις, είτε να επιδιώξουμε λύση στα πλαίσια της μεθόδου “*Εύρεση Προσεγγιστικών Ιδιοδιανυσμάτων*”.

Στον κώδικα διαμόρφωσης που κατασκευάσαμε επιλέξαμε να εργαστούμε με τη δεύτερη επιλογή, την οποία και παρουσιάζουμε εκτενώς παρακάτω.

Εύρεση Προσεγγιστικών Ιδιοδιανυσμάτων

Με τη μέθοδο της Εύρεσης Προσεγγιστικών Ιδιοδιανυσμάτων μας δίνεται η δυνατότητα να φράξουμε κάτωθεν, με ένα κοινό, για όλες τις καταστάσεις όριο, το πλήθος των κωδικών λέξεων, που κάθε κατάσταση διαθέτει.

Πιο συγκεκριμένα αν συμβολίσουμε τον πίνακα A^{16} , διαστάσεων 11×11 , με T και με M το ελάχιστο πλήθος λέξεων κάθε κατάσταση. Μπορούμε να θεωρήσουμε το σύστημα $T \cdot u \geq M \cdot u$, με το u να είναι το διάνυσμα 11×1 , που λύνει το σύστημα.

Ήδη αναφέραμε πως το M δηλώνει το ελάχιστο πλήθος των κωδικών λέξεων που μπορεί να έχει κάθε κατάσταση και ως τέτοιο είναι ένας θετικός ακέραιος αριθμός. Προφανώς ως κάτω όριο τίθεται πάλι το M να είναι ίσο του 256, με την επιθυμία μας να είναι το M να λάβει όσο το δυνατόν μεγαλύτερη τιμή γίνεται. Το 11×1 διάνυσμα u λαμβάνει θετικά ορισμένες, ακέραιες τιμές. Σε αντίθεση με το M , στην περίπτωση του u , ενδιαφερόμαστε για την ελαχιστοποίηση των τιμών των στοιχείων του. Οι λόγοι που κάνουν προτιμητέα τη μεγιστοποίησή από τη μία και την ελαχιστοποίηση από την άλλη θα εξηγηθούν αναλυτικά σε επόμενο σημείο της εργασίας.

Το προς επίλυση πλέον σύστημα διατυπώνεται ως εξής:

$$T \cdot u \geq M \cdot u$$

Το T , που συμβολίζει τον πίνακα A^{16} , είναι το μοναδικό γνωστό δεδομένο του προβλήματος. Αναζητούμε ένα θετικά ορισμένο, ακέραιο διάνυσμα u , διαστάσεων 11×1 , με τα στοιχεία του να λαμβάνουν όσο το δυνατόν μικρότερες τιμές και συνολικά να μας δίνουν όσο το δυνατόν μεγαλύτερο συντελεστή M . Πρόκειται για

ένα δύσκολο πρόβλημα βελτιστοποίησης, το οποίο ανήκει στην περιοχή του ακέραιου προγραμματισμού.

Ο πρώτος που ασχολήθηκε με την εύρεση ενός αποδοτικού αλγορίθμου επίλυσης του παραπάνω συστήματος ήταν ο Peter A. Franaszek. Στο [8] υπάρχει η αναλυτική λύση που ο ίδιος πρότεινε και δημοσίευσε στο ερευνητικό περιοδικό της IBM το 1980. Στο παρακάτω πλαίσιο δίνεται ο αλγόριθμος.

Αλγόριθμος Προσέγγισης Ιδιοδιάνυσματος Franaszek

[Βήμα 1]: Όρισε κατάλληλη μεταβλητή K και αρχικοποίησε την με την τιμή 0.

[Βήμα 2]: Όρισε το διάνυσμα $u^{(0)}$. Αρχικοποίησε όλα του τα στοιχεία με τη μέγιστη επιτρεπόμενη τιμή διανύσματος L . $u^{(0)} = \{L, L, \dots, L\}$

[Βήμα 3]: Όρισε το νέο διάνυσμα $u^{(k+1)}$ ως εξής. Κάθε στοιχείο $u^{(k+1)} [i]$ του διανύσματος $u^{(k+1)}$ υπολογίζεται από τη σχέση:

$$\min(u^{(k)} [i], (1/n) * \sum_j (t[i][j] * u^{(k)} [j]))$$

[Βήμα 4]: Εάν $u^{(k+1)} \neq u^{(k)}$. Αύξησε το k κατά 1 και πήγαινε στο βήμα 3.

[Βήμα 5]: Εάν $u^{(k+1)} == u^{(k)}$. Θέσε το u ίσο με $u^{(k)}$ και τερμάτισε.

Στον πίνακα, που ακολουθεί, φαίνονται τα αποτελέσματα έτσι όπως παράγονται από την εκτέλεση του αλγορίθμου Franaszek.

Πίνακας Αποτελεσμάτων Αλγορίθμου Franaszek

M	Μέγιστο Στοιχείο Ιδιοδιανύσματος
351	2
353	3
385	4
389	8
391	10
397	13
398	17
400	19
402	23
403	32
404	47
405	51
406	102

Ο κώδικας διαμόρφωσης καναλιού EFMplus βασίζει τη σχεδίαση του σε **M=351**, με το αντίστοιχο διάνυσμα να υπολογίζεται από τον παραπάνω αλγόριθμο στο:

$$u = [1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 0], M=351$$

Ο κώδικας διαμόρφωσης καναλιού, που εμείς σχεδιάσαμε, βασίζεται σε **M=385**, με το αντίστοιχο διάνυσμα να είναι:

$$u = [2, 3, 4, 4, 4, 4, 3, 3, 3, 2, 1], M=385$$

Η επιλογή μας να εργαστούμε με τη λύση της «Εύρεσης Προσεγγιστικών Ιδιοδιανυσμάτων» μας “αναγκάζει” να επαναλάβουμε τους θεωρητικούς υπολογισμούς ως προς την αποδοτικότητα, που παρουσιάσαμε παραπάνω. Πλέον στον υπολογισμό της αποδοτικότητας θα συμμετάσχει ο συντελεστής M. Το M συμβολίζει το ελάχιστο πλήθος των λέξεων κάθε κατάσταση του κώδικα διαμόρφωσης. Το πλήθος των λέξεων μας ενδιαφέρει να είναι όσο το δυνατόν μεγαλύτερο για λόγους που θα γίνουν γνωστοί παρακάτω. Ο τύπος, που δώσαμε προηγουμένως, θα διαμορφωθεί στον:

$$\eta = R/C = (\log_2 M/q)/C$$

Η αποδοτικότητα πλέον του EFMplus υπολογίζεται σε 0.9754, ενώ αντίστοιχα του δικού μας κώδικα σε 0.9908. Η αποδοτικότητα μας δείχνει πόσο “κοντά” βρίσκεται ο ρυθμός κωδικοποίησης στη χωρητικότητα.

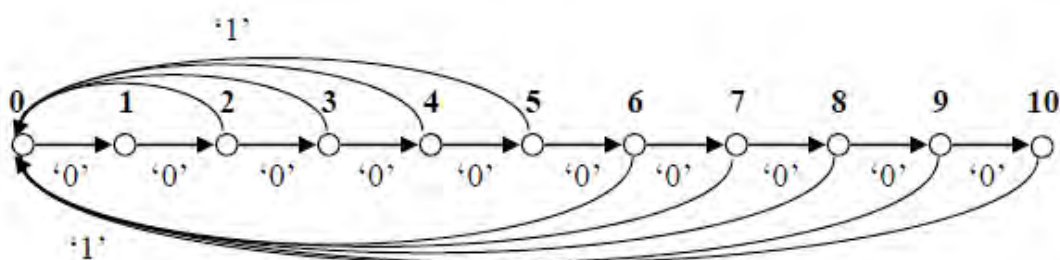
Συγκώνευση Καταστάσεων

Σημειώσαμε παραπάνω πως ο δικός μας κώδικας διαμόρφωσης καναλιού βασίζει τη σχεδίαση του σε **M=385**, με το αντίστοιχο διάνυσμα είναι:

$$u = [2, 3, 4, 4, 4, 4, 3, 3, 3, 2, 1], M=385$$

Το παραπάνω 11X1 διάνυσμα ουσιαστικά μας σημειώνει την ύπαρξη έντεκα διαφορετικών καταστάσεων. Θα ονομάζουμε κάθε έναν συντελεστή από τους έντεκα συντελεστές του διανύσματος ως “βάρος”. Η κατάσταση 0 έχει βάρος 2, η κατάσταση 1 σε βάρος 3 και ούτω καθ’ εξής. Κάθε βάρος διάφορο της μονάδας($\neq 1$), μας υποδεικνύει το διαχωρισμό της αντίστοιχης κατάστασης, σε νέες καταστάσεις. Έστω κατάσταση A με βάρος X, όπου $X \neq 1$, τότε στο βιβλίο κωδικών στη θέση της κατάστασης A θα πρέπει να δημιουργηθούν X νέες καταστάσεις. Δηλαδή η κατάσταση A θα πρέπει να διαχωριστεί σε X νέες καταστάσεις. Γίνεται πλέον τώρα σαφές, γιατί κατά την επίλυση του συστήματος $T*u \geq M*u$, μας ενδιέφερε τα στοιχεία του διανύσματος u να λαμβάνουν όσο το δυνατόν μικρότερες τιμές. Μεγάλες τιμές στο διάνυσμα u θα απαιτούσαν διαχωρισμούς καταστάσεων σε πολλές νέες καταστάσεις, κάτι που θα δυσκόλευε σημαντικά το έργο της σχεδίασης του διαμορφωτή.

Πριν προχωρήσουμε στο διαχωρισμό καταστάσεων θα επιχειρήσουμε τη συγχώνευση των καταστάσεων εκείνων που έχουν το ίδιο βάρος. Το κοινό βάρος σε δύο ή περισσότερες καταστάσεις δεν μας εγγυάται τη συγχώνευση τους, ωστόσο την κάνει αρκετά πιθανή. Έτσι λοιπόν οι καταστάσεις 3, 4, 5 και 6, που έχουν βάρος 4 πιθανόν να μπορούν να συγχωνευτούν σε μία κατάσταση. Με αντίστοιχο τρόπο αντιμετωπίζονται και οι καταστάσεις 6, 7 και 8 βάρους 3.



Σε αυτό το σημείο θα διακόψουμε τη συζήτηση μας για να ορίσουμε την έννοια του “Follower Set”. Παρατηρώντας τον παραπάνω γράφο συμπεραίνουμε εύκολα ότι κάθε κατάσταση του κώδικα διαμόρφωσης μπορεί να περιγραφεί πλήρως από το πλήθος των τελικών μηδενικών (*number of trailing zeros*) που έχουν οι κωδικές λέξεις που καταλήγουν σε αυτήν. Για παράδειγμα στην κατάσταση 0 είναι δυνατόν να καταλήξουν μόνο κωδικές λέξεις με πλήθος τελικών μηδενικών (*number of trailing zeros*) ίσο με μηδέν. Άρα αντιστρέφοντας τον παραπάνω συλλογισμό σημειώνουμε πως η κατάσταση 0 περιλαμβάνει όλες εκείνες τις κωδικές λέξεις, που ξεκινάνε με τουλάχιστον 2 και το πολύ μέχρι 10 μηδενικά (*αριθμός αρχικών μηδενικών – number of leading zeros*). Ονομάζουμε το τελευταίο σύνολο “Follower Set” και τούδε και στο εξής θα τον χρησιμοποιούμε κάθε φορά που θέλουμε να περιγράψουμε μια κατάσταση. Στον παρακάτω πίνακα φαίνεται το “Follower Set” έτσι όπως υπολογίστηκε για κάθε κατάσταση.

Κατάσταση	"Follower Set"	Βάρος
0	{2,...,10}	2
1	{1,...,9}	3
2	{0,...,8}	4
3	{0,...,7}	4
4	{0,...,6}	4
5	{0,...,5}	4
6	{0,...,4}	3
7	{0,...,3}	3
8	{0,...,2}	3
9	{0,...,1}	2
10	{0,...,0}	1

Όπως έχουμε ήδη αναφέρει οι κωδικές λέξεις που μας ενδιαφέρουν είναι δυαδικές, μήκους 16 λέξεις, που υπακούουν στον περιορισμό $(d, k) = (2, 10)$. Το πλήθος αυτών ισούται με 566. Χρησιμοποιώντας ως κριτήριο το πλήθος των αρχικών μηδενικών (*number of leading zeros*) και το πλήθος των τελικών μηδενικών (*number of trailing zeros*) κάθε λέξης, ταξινομούμε αυτές τις λέξεις στον πίνακα που παρουσιάζεται παρακάτω. Όπου για παράδειγμα, το στοιχείο στη θέση [0][3] του πίνακα, που ισούται με 27, ουσιαστικά να μας πληροφορεί πως στο σύνολο των 566 λέξεων περιλαμβάνονται ακριβώς 27 λέξεις με πλήθος αρχικών μηδενικών ίσο με 3 και με πλήθος τελικών μηδενικών ίσο με 0.

57,	40,	27,	18,	13,	9,	6,	4,	3,	2,	1
40,	27,	18,	13,	9,	6,	4,	3,	2,	1,	1
27,	18,	13,	9,	6,	4,	3,	2,	1,	1,	1
18,	13,	9,	6,	4,	3,	2,	1,	1,	1,	0
13,	9,	6,	4,	3,	2,	1,	1,	1,	0,	0
9,	6,	4,	3,	2,	1,	1,	1,	0,	0,	1
6,	4,	3,	2,	1,	1,	1,	0,	0,	1,	0
4,	3,	2,	1,	1,	1,	0,	0,	1,	0,	0
3,	2,	1,	1,	1,	0,	0,	1,	0,	0,	0
2,	1,	1,	1,	0,	0,	1,	0,	0,	0,	0
1,	1,	1,	0,	0,	1,	0,	0,	0,	0,	0

Παρατηρούμε πως ο παραπάνω πίνακας είναι συμμετρικός. Για να εξηγήσουμε αυτή του συμμετρία θα δώσουμε το παρακάτω παράδειγμα. Έστω η έγκυρη λέξη 0x0202, με δυαδική αναπαράσταση 0000001000000010. Εάν θεωρήσουμε τη σύμβαση πως τα αριστερότερα ψηφία είναι τα "αρχικά ψηφία" της λέξης και τα αντίστοιχα δεξιότερα τα "τελικά ψηφία" της λέξης, μπορούμε ορθά να πούμε πως το πλήθος των αρχικών μηδενικών της λέξης 0x0202 ισούται με 6 και το αντίστοιχο των τελικών με 1. Προφανώς και η αντίστροφη λέξη, δηλαδή η 0x4040 με δυαδική αναπαράσταση 0100000001000000, είναι επίσης επιτρεπτή. Η λέξη έχει πλήθος αρχικών μηδενικών ίσο με 1 και αντίστοιχο τελικών με 6. Η παραπάνω συμμετρία, που παρουσιάζουν οι λέξεις που μελετάμε, είναι αυτή που δίνει αυτή τη συμμετρική μορφή στον παραπάνω πίνακα.

Με τη βοήθεια του πίνακα, που μόλις παρουσιάσαμε, υπολογίσαμε με ακρίβεια το “Follower Set” κάθε κατάστασης. Δείξαμε λοιπόν πως οι καταστάσεις 5, 4, 3 και 2 είναι η μία υποσύνολο της άλλης κάτι που επιτρέπει τη συγχώνευση τους σε μία ενιαία κατάσταση με “Follower Set” {0,...,5}. Αντίστοιχα οι καταστάσεις 8, 7 και 6 συγχωνεύονται σε μία αντίστοιχη ενιαία με “Follower Set” {0,...,2}. Πλέον οι καταστάσεις και τα “Follower Sets” έχουν διαμορφωθεί ως εξής:

<u>Κατάσταση</u>	<u>“Follower Set”</u>	<u>Βάρος</u>	<u>Πλήθη</u>
0	{2,...,10}	2	794
1	{1,...,9}	3	1156
2	{0,...,5}	4	1560
3	{0,...,2}	3	1178
4	{0,...,1}	2	921
5	{0,...,0}	1	546

Με την κατάσταση [0] να πρέπει να διαχωριστεί σε δύο καταστάσεις, την [1] σε τρεις, την [2] σε τέσσερις, την [3] σε τρεις, την [4] σε δύο και την [1] να μένει ως έχει. Άρα μετά την σωστή εφαρμογή της τεχνικής του διαχωρισμού καταστάσεων θα έχουμε στη διάθεση μας έναν διαμορφωτή δεκαπέντε καταστάσεων.

Διαχωρισμός Καταστάσεων

Στα πλαίσια αυτής της εργασίας και στην προσπάθεια μας να οδηγηθούμε σε βέλτιστο αποτέλεσμα δοκιμάσαμε αρκετές μεθόδους διαχωρισμού καταστάσεων. Από τις σαράντα περίπου διαφορετικές μεθόδους εμείς ξεχωρίσαμε και παρουσιάζουμε τρεις μόνο από αυτές. Για κάθε μέθοδο σημειώνουμε τα κριτήρια διαχωρισμού που χρησιμοποιήθηκαν, καθώς και τα σχετικά πλήθη λέξεων, έτσι όπως αυτά προέκυψαν, μετά το διαχωρισμό.

> Μέθοδος 1:

Τυχαίος Διαχωρισμός: Στη “Μέθοδο 1” προχωρήσαμε στο διαχωρισμό των καταστάσεων με εντελώς τυχαία κριτήρια. Μοναδικό μας μέλημα υπήρξε η όσο το δυνατόν πιο ισομερής κατανομή των λέξεων στις νέες καταστάσεις.

Πλήθη Καταστάσεων:

Κατάσταση 0: 398 και 396

Κατάσταση 1: 385, 386 και 385

Κατάσταση 2: 370, 425, 383, 382

Κατάσταση 3: 378, 419 και 381

Κατάσταση 4: 434 και 487

Κατάσταση 5: 512

*Η Κατάσταση 5 διαθέτει 546 λέξεις. 512 από αυτές επιλέχτηκαν τυχαία, ώστε να συμπεριληφθούν στο βιβλίο κωδικών.

> Μέθοδος 2:

Κριτήρια:

Κατάσταση 0: Ισοτιμία

Κατάσταση 1: Ισοτιμία και $((NLZ \geq 2) \ || \ ((NLZ == 0) \ \&\& \ (NTZ \geq 4)))$

Κατάσταση 2: Ισοτιμία και $((NLZ \geq 2) \ || \ ((NLZ == 0) \ \&\& \ (NTZ \geq 4)))$

Κατάσταση 3: Ισοτιμία και $((NLZ \geq 2) \ || \ ((NLZ == 0) \ \&\& \ (NTZ \geq 4)))$

Κατάσταση 4: Ισοτιμία και $((NLZ \geq 2) \ || \ ((NLZ == 0) \ \&\& \ (NTZ \geq 4)))$

Κατάσταση 5: Δεν εφαρμόστηκε κανένα κριτήριο.

Πλήθη Καταστάσεων:

Κατάσταση 0: 405 και 389

Κατάσταση 1: 401, 380 και 375

Κατάσταση 2: 409, 362, 377 και 412

Κατάσταση 3: 389, 377 και 412

Κατάσταση 4: 450 και 471

Κατάσταση 5: 471

> Μέθοδος 3: Η “Μέθοδος 3” βασίστηκε στο διαχωρισμό της “Μεθόδου 2”, με μοναδική διαφορά την ενσωμάτωση 41 επιπλέον λέξεων στη κατάσταση 15.

Κριτήρια:

Κατάσταση 0: Ισοτιμία

Κατάσταση 1: Ισοτιμία και $((NLZ \geq 2) \ || \ ((NLZ == 0) \ \&\& \ (NTZ \geq 4)))$

Κατάσταση 2: Ισοτιμία και $((NLZ \geq 2) \ || \ ((NLZ == 0) \ \&\& \ (NTZ \geq 4)))$

Κατάσταση 3: Ισοτιμία και $((NLZ \geq 2) \ || \ ((NLZ == 0) \ \&\& \ (NTZ \geq 4)))$

Κατάσταση 4: Ισοτιμία και $((NLZ \geq 2) \ || \ ((NLZ == 0) \ \&\& \ (NTZ \geq 4)))$

Κατάσταση 5: Δεν εφαρμόστηκε κανένα κριτήριο.

Πλήθη Καταστάσεων:

Κατάσταση 0: 405 και 389

Κατάσταση 1: 401, 380 και 375

Κατάσταση 2: 409, 362, 377 και 412

Κατάσταση 3: 389, 377 και 412

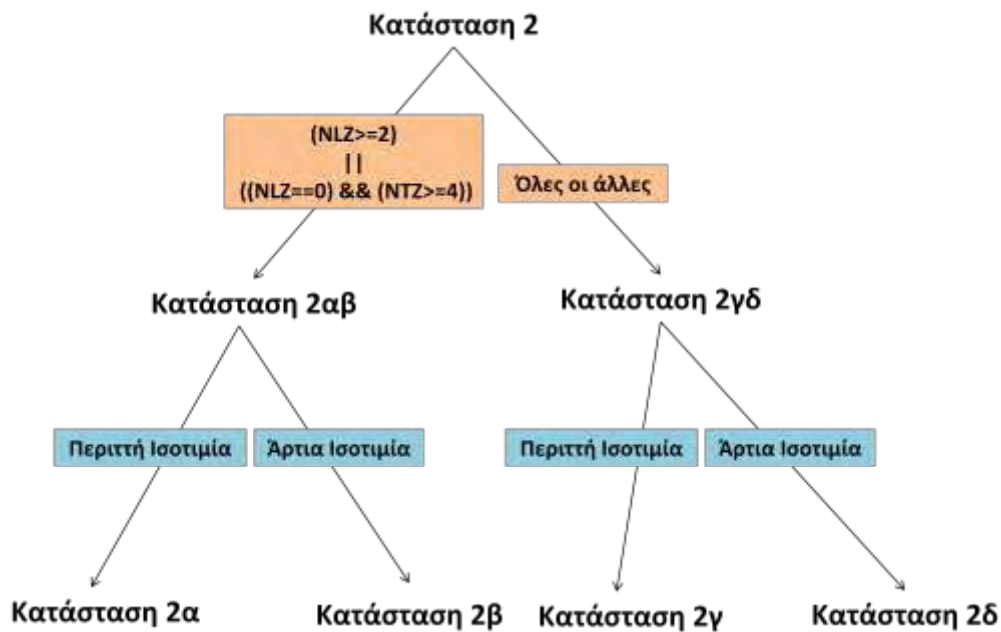
Κατάσταση 4: 450 και 471

Κατάσταση 5: 512

Παραπάνω με *NLZ* συμβολίζουμε το πλήθος των αρχικών μηδενικών (*number of leading zeros*) της λέξης και με *NTZ* συμβολίζουμε το αντίστοιχο των τελικών μηδενικών (*number of trailing zeros*).

Στην προσπάθεια μας να γίνουν όσο το δυνατόν πιο κατανοητές οι μέθοδοι διαχωρισμού θα παρουσιάσουμε εκτενώς το διαχωρισμό της κατάστασης [2] σε 4 νέες καταστάσεις. Επιλέξαμε να χρησιμοποιήσουμε το σύνολο κριτηρίων της μεθόδου [2]. Η κατάσταση [2] διαθέτει 1560 λέξεις και έχει βάρος ίσο με 4. Άρα πρέπει να διαχωριστεί σε 4 νέες καταστάσεις. Ας θεωρήσουμε πως στην κατάσταση [2] ανήκει η λέξη 0x2411, με δυαδική αναπαράσταση 0010010000010001. Η λέξη έχει αριθμό αρχικών μηδενικών (*NLZ*) ίσο με 2 και αντίστοιχο τελικών (*NTZ*) ίσο με 0. Κατ’ αντίστοιχο τρόπο η λέξη 0x8122, με δυαδική αναπαράσταση 1000000100100010, έχει *NLZ* ίσο με 0 και *NTZ* ίσο με 1. Αν τώρα θεωρήσω το κριτήριο $((NLZ \geq 2) \ || \ ((NLZ == 0) \ \&\& \ (NTZ \geq 4)))$, τότε η λέξη 0x2411, που ικανοποιεί το κριτήριο, θα τοποθετηθεί σε μία α’ ομάδα, ενώ η λέξη 0x8122, που δεν το ικανοποιεί, θα τοποθετηθεί σε μία β’ ομάδα. Αν επαναλάβω το παραπάνω για όλες

τις λέξεις της κατάστασης [2], τότε μετά την εξέταση όλων των λέξεων θα έχω καταφέρει να διαχωρίσω την κατάσταση σε δύο νέες καταστάσεις, στην κατάσταση Κατάσταση [2αβ] και Κατάσταση [2γδ]. Η Κατάσταση [2αβ] περιέχει όλες τις λέξεις που ικανοποιούν το κριτήριο που ορίστηκε παραπάνω, ενώ η [2γδ] όλες τις υπόλοιπες, δηλαδή όλες όσες δεν το ικανοποιούν. Έστω τώρα οι λέξεις 0x1024 και 0x2411 της Κατάστασης [2αβ], με αντίστοιχες δυαδικές αναπαραστάσεις 0001000000100100 και 0010010000010001. Παρατηρούμε πως η πρώτη λέξη έχει 3 άσσους στη σύνθεση της, ενώ η δεύτερη 4. Μπορούμε λοιπόν να πούμε πως η 0x1024 παρουσιάζει περιττή ισοτιμία και η 0x2411 αντίστοιχη άρτια. Αν τοποθετήσουμε όλες τις λέξεις με περιττή ισοτιμία σε μία Κατάσταση [2α] και όλες τις λέξεις με αντίστοιχη άρτια σε μια Κατάσταση [2β], τότε θα έχουμε επιτυχώς διαχωρίσει την Κατάσταση [2αβ] σε δύο νέες καταστάσεις. Με ακριβώς τον ίδιο τρόπο διασπούμε την Κατάσταση [2γδ] στις καταστάσεις [2γ] και [2δ]. Παρακάτω παρουσιάζεται γραφικά ο διαχωρισμός της Κατάστασης [2] στις τέσσερις νέες καταστάσεις.



Στη βιβλιογραφία μεθόδους διαχωρισμού, όπου επιλέγουμε να διαχωρίζουμε τις λέξεις μιας κατάστασης βασιζόμενοι στα χαρακτηριστικά των λέξεων της, για παράδειγμα περιττή ή άρτια ισοτιμία, συναντώνται ως Συστηματικές Μέθοδοι Διαχωρισμού(*Systematic Splitting Methods*).

Η μέθοδος διαχωρισμού [2] προτάθηκε από τη Μάρθα Πέππα. Η μέθοδος [3] βασίζεται σε μεγάλο βαθμό στη μέθοδο [2] και προτάθηκε από εμένα. Στη συνέχεια της εργασίας θα αναφερθούμε στην υπεροχή της μεθόδου [2] συγκριτικά τόσο με τη [3], όσο και με την μέθοδο τυχαίου διαχωρισμού[1].

Συνεπής Ανάθεση Κωδικών Λέξεων σε Λέξεις Εισόδου

Η κωδικοποίηση στον κώδικα διαμόρφωσης που κατασκευάσαμε βασίζεται σε έναν διαμορφωτή δεκαπέντε καταστάσεων. Μπορούμε να φανταστούμε για αρχή αυτόν τον κωδικοποιητή ως μία κλασική Μηχανή Πεπερασμένων Καταστάσεων(*Finite*

State Machine). Κατά αντίστοιχο τρόπο ο αποκωδικοποιητής είναι άμεσα εξαρτημένος από τις καταστάσεις, έτσι όπως αυτές προέκυψαν από τη διαδικασία της κωδικοποίησης, με την ακριβή αποκωδικοποίηση να προϋποθέτει “βάδισμα” επάνω σε αυτές.

Ενδεχομένως σε ένα ιδανικό περιβάλλον απουσίας θορύβου, η εξάρτηση της αποκωδικοποίησης από τις καταστάσεις να μην άξιζε της προσοχής μας, ωστόσο σε ένα πραγματικό περιβάλλον, που η παρουσία θορύβου είναι κάτι παραπάνω από σίγουρη, η χρήση αποκωδικοποιητή εξαρτώμενου από καταστάσεις μπορεί να οδηγήσει σε ανακριβή αποκωδικοποίηση. Ιδανικά θα θέλαμε έναν αποκωδικοποιητή εντελώς ανεξάρτητο από τις καταστάσεις.

Πριν προχωρήσουμε άλλο τη συζήτηση μας θα διευκρινίσουμε την ορολογία μας. Ως κωδική λέξη θεωρούμε το ζευγάρι λέξη και επόμενη κατάσταση. Για παράδειγμα το ζευγάρι 0x1220, 3 είναι μία κωδική λέξη. Από εδώ και στο εξής για να κάνουμε αναφορά, για παράδειγμα μόνο στο 0x1220, θα χρησιμοποιούμε τον όρο λέξη κωδικής λέξης και για αναφορά στο 3 κατάσταση κωδικής λέξης. Η κωδική λέξη 0x1220, 3 κωδικοποιεί τη λέξη 89, θα αναφερόμαστε στη 89 ως λέξη χρήστη.

Ας υποθέσουμε πως έχουμε το παρακάτω μικρό απόσπασμα από το βιβλίο κωδικών ενός κώδικα διαμόρφωσης τεσσάρων καταστάσεων.

```
/* 89 */ 0x1220, 3, 0x2108, 3, 0x4109, 1, 0x4489, 1,  
/* 90 */ 0x2010, 3, 0x2010, 3, 0x4249, 1, 0x4249, 1,  
/* 91 */ 0x2084, 3, 0x1220, 3, 0x4489, 1, 0x4109, 1,  
/* 92 */ 0x2108, 3, 0x4901, 1, 0x2084, 3, 0x4901, 1,
```

Έστω ότι κατά την αποδιαμόρφωση, θέλουμε να αποκωδικοποιήσουμε τη κωδική λέξη 0x1220, 3. Η συγκεκριμένη κωδική λέξη βρίσκεται στην κατάσταση 1, κωδικοποιώντας τη λέξη χρήστη 89, και στην κατάσταση 2, όπου κωδικοποιεί τη λέξη χρήστη 91. Άρα η αποκωδικοποίηση εξαρτάται αποκλειστικά από την σωστή εκτίμηση της κατάστασης στην οποία βρισκόμαστε. Αν υποθέσουμε πως η κατάσταση που βρισκόμαστε είναι η 1, αλλά λόγω λαθμενής εκτίμησης θεωρήσουμε πως βρισκόμαστε στην κατάσταση 2, αυτό θα έχει ως συνέπεια η κωδική λέξη 0x1220, 3 να αποκωδικοποιηθεί εσφαλμένα στη λέξη χρήστη 91. Το πρόβλημα αυτό θα μπορούσε να εξαλειφτεί, εάν κατά την κατασκευή του βιβλίου κωδικών αναθέταμε ίδιες κωδικές λέξεις σε ίδιες λέξεις χρήστη. Πρακτικά, αν τοποθετούσαμε ίδιες κωδικές λέξεις στην ίδια γραμμή, θα είχαμε έναν αποκωδικοποιητή σε μεγάλο βαθμό ανεξάρτητο από τις καταστάσεις. Σε αυτήν την περίπτωση το παραπάνω απόσπασμα του βιβλίου κωδικών θα γινόταν:

```
/* 89 */ 0x1220, 3, 0x1220, 3, 0x4109, 1, 0x4109, 1,  
/* 90 */ 0x2010, 3, 0x2010, 3, 0x4249, 1, 0x4249, 1,  
/* 91 */ 0x2084, 3, 0x2084, 3, 0x4489, 1, 0x4489, 1,  
/* 92 */ 0x2108, 3, 0x4901, 1, 0x2108, 3, 0x4901, 1,
```

Όταν δύο ή περισσότερες καταστάσεις είναι αποτέλεσμα διαχωρισμού(*splitting*), τότε η ανάθεση κωδικών λέξεων πρέπει να γίνει με τέτοιο τρόπο, ώστε αποκωδικοποίηση να παραμείνει όσο το δυνατόν πιο ανεξάρτητη από τις καταστάσεις. Πρακτικά για να αποφύγουμε το πρόβλημα που περιγράψαμε και παραπάνω τοποθετούμε ίδιες λέξεις κωδικής λέξης στην ίδια κατάσταση.

Η τοποθέτηση των κωδικών λέξεων στο βιβλίο κωδικών με τον τρόπο που μόλις περιγράψαμε, συναντάται στη βιβλιογραφία με τον όρο συνεπής ανάθεση(*consistent assignment* ή *consistent naming*). Θα μπορούσαμε να συνοψίσουμε την έννοια της συνεπούς ανάθεσης στις τρεις παρακάτω συνθήκες, με το βιβλίο κωδικών να είναι να θεωρείται συνεπές(*consistent*) μόνο όταν και οι τρεις συνθήκες είναι αληθείς.

Συνθήκη 1: Έστω ότι υπάρχει κωδική λέξη A, η οποία ανήκει σε μια κατάσταση B του βιβλίου κωδικών και κωδικοποιεί τη λέξη χρήστη Γ. Εάν υπάρχει η A σε οποιαδήποτε κατάσταση, διάφορη του B($\neq B$), του βιβλίου κωδικών και κωδικοποιεί λέξη χρήστη διάφορη του Γ($\neq \Gamma$). Τότε το βιβλίο κωδικών ΔΕΝ είναι συνεπές.

Παράδειγμα μη συνεπούς βιβλίου κωδικών:

```
/* 89 */ Ox1220, 3, 0x2010, 3, 0x4109, 1, 0x4109, 1,  
/* 90 */ 0x2010, 3, Ox1220, 3, 0x4249, 1, 0x4249, 1,  
/* 91 */ 0x2084, 3, 0x2084, 3, 0x4489, 1, 0x4489, 1,  
/* 92 */ 0x2108, 3, 0x4901, 1, 0x2108, 3, 0x4901, 1,
```

Παράδειγμα συνεπούς βιβλίου κωδικών:

```
/* 89 */ Ox1220, 3, Ox1220, 3, 0x4109, 1, 0x4109, 1,  
/* 90 */ 0x2010, 3, 0x2010, 3, 0x4249, 1, 0x4249, 1,  
/* 91 */ 0x2084, 3, 0x2084, 3, 0x4489, 1, 0x4489, 1,  
/* 92 */ 0x2108, 3, 0x4901, 1, 0x2108, 3, 0x4901, 1,
```

Συνθήκη 2: Έστω ότι υπάρχει βιβλίο κωδικών αποτελούμενου από τέσσερις καταστάσεις(E, Z, H και Θ), με τις καταστάσεις E και Z να είναι αποτέλεσμα διαχωρισμού(*splitting*). Ας υποθέσουμε τώρα πως υπάρχει κωδική λέξη A, η οποία ανήκει στην Z με λέξη κωδικής λέξης B. Εάν υπάρχει η λέξη κωδικής λέξης B στη κατάσταση E, και ανεξάρτητα από το ποια λέξη χρήστη κωδικοποιεί, τότε το βιβλίο κωδικών ΔΕΝ είναι συνεπές.

Παράδειγμα μη συνεπούς βιβλίου κωδικών:

```
/* 18 */ 0x2402, 3, 0x0881, 1, 0x4109, 1, 0x2402, 3,  
/* 19 */ 0x2492, 3, 0x0881, 2, 0x4109, 2, 0x2492, 3,  
/* 20 */ 0x2024, 3, Ox0901, 1, 0x4111, 1, 0x2024, 3,  
/* 21 */ Ox0901, 3, 0x2044, 2, 0x4111, 2, 0x2044, 3,
```

Παράδειγμα συνεπούς βιβλίου κωδικών:

```
/* 18 */ 0x2402, 3, 0x0881, 1, 0x4109, 1, 0x2402, 3,  
/* 19 */ 0x2492, 3, 0x0881, 2, 0x4109, 2, 0x2492, 3,  
/* 20 */ 0x2024, 3, Ox0901, 1, 0x4111, 1, 0x2024, 3,  
/* 21 */ 0x2044, 3, Ox0901, 2, 0x4111, 2, 0x2044, 3,
```


***Υποθέτω πως οι καταστάσεις 1 και 2 είναι αποτέλεσμα διαχωρισμού**

Συνθήκη 3: Το πλήθος των διαθέσιμων κωδικών λέξεων για την περίπτωση που μελετάμε δεν επαρκεί για ανάθεση κωδικής λέξης σε όλες τις λέξεις χρήστη, για όλες τις καταστάσεις, και στο κύριο βιβλίο κωδικών(*main codebook*) και στο εναλλακτικό βιβλίο κωδικών(*alternate codebook*). Το πρόβλημα αυτό το επιλύουμε τοποθετώντας 0xFFFF, 0, κάθε φορά που δεν έχουμε διαθέσιμη κωδική λέξη. Εάν υπάρχει λέξη χρήστη A, που κωδικοποιείται σε 0xFFFF, 0 στην κατάσταση B του κύριου βιβλίου κωδικών, και η ίδια λέξη χρήστη κωδικοποιείται επίσης σε 0xFFFF, 0 στην κατάσταση B του εναλλακτικού βιβλίου κωδικών, τότε το βιβλίο κωδικών ΔΕΝ είναι συνεπές.

Παράδειγμα μη συνεπούς βιβλίου κωδικών:

Κύριο/Main:

```
/* 18 */ 0x2402, 3, 0x0881, 1, 0x4109, 1, 0x2402, 3,  
/* 19 */ 0x2492, 3, 0xFFFF, 0, 0x4109, 2, 0x2492, 3,
```

Εναλλακτικό/Alternate:

```
/* 18 */ 0x2024, 3, 0x0901, 1, 0x4111, 1, 0x2024, 3,  
/* 19 */ 0x2044, 3, 0xFFFF, 0, 0x4111, 2, 0x2044, 3,
```

Παράδειγμα συνεπούς βιβλίου κωδικών:

Κύριο/Main:

```
/* 18 */ 0x2402, 3, 0x0881, 1, 0x4109, 1, 0x2402, 3,  
/* 19 */ 0x2492, 3, 0x0881, 2, 0x4109, 2, 0x2492, 3,
```

Εναλλακτικό/Alternate:

```
/* 18 */ 0x2024, 3, 0x0901, 1, 0x4111, 1, 0x2024, 3,  
/* 19 */ 0x2044, 3, 0xFFFF, 0, 0x4111, 2, 0x2044, 3,
```

Διαδικασία Διαμόρφωσης/Κωδικοποίησης

Στα εισαγωγικά κεφάλαια της εργασίας αναφερθήκαμε εκτενώς στην RDS διασπορά, εξηγήσαμε τη σημασία της και τονίσαμε το ενδιαφέρον μας για την ελαχιστοποίησή της. Η εξασφάλιση της ελάχιστης δυνατής RDS διασποράς είναι και το μοναδικό μας μέλημα κατά τη διαμόρφωση/κωδικοποίηση ενός αρχείου. Η σχετική διαδικασία θα επιδειχθεί παρακάτω με τη βοήθεια ενός παραδείγματος.

Έστω λοιπόν ένα βιβλίο κωδικών με τέσσερις καταστάσεις, μέρη του οποίου φαίνονται παρακάτω.

```
...  
/* 19 */ 0x2091, 1, 0x2091, 1, 0x8880, 4, 0x8880, 4,  
/* 20 */ 0x2121, 1, 0x2121, 1, 0x9100, 4, 0x9100, 4,  
/* 21 */ 0x2241, 1, 0x2241, 1, 0x4049, 1, 0x4049, 1,  
/* 22 */ 0x2481, 1, 0x2481, 1, 0x8021, 1, 0x8021, 1,  
/* 23 */ 0x1004, 2, 0x1004, 2, 0x4088, 2, 0x4088, 2,  
/* 24 */ 0x1024, 2, 0x1024, 2, 0x4088, 3, 0x4088, 3,  
...  
/* 19 */ 0x0110, 2, 0x4920, 3, 0x8110, 2, 0x4920, 3,  
/* 20 */ 0x0088, 2, 0x4824, 2, 0x8088, 2, 0x4824, 2,
```

/* 21 */ 0x0044, 2, 0x4812, 1, 0x8044, 2, 0x4812, 1,
 /* 22 */ 0x2401, 1, 0x2401, 1, 0x9024, 3, 0x9024, 3,
 /* 23 */ 0x1201, 1, 0x1201, 1, 0x9004, 3, 0x9004, 3,
 /* 24 */ 0x0901, 1, 0x0901, 1, 0x9120, 2, 0x9120, 2,

...

Το κύριο βιβλίο κωδικών(*main codebook*) χρωματίζεται με μαύρο, ενώ το εναλλακτικό βιβλίο κωδικών(*alternate codebook*) με πορτοκαλί.

Ας υποθέσουμε χάριν παραδείγματος, πως τα δύο πρώτα *bytes* του υπό διαμόρφωση αρχείου είναι:

Σε Δυαδική Αναπαράσταση: **00010011** και **00011000**

Σε Δεκαδική Αναπαράσταση: **19** και **24**

Σε Δεκαεξαδική Αναπαράσταση: **0x13** και **0x18**

Ο διαμορφωτής/κωδικοποιητής κατά τα γνωστά θα λαμβάνει *8 user bits* και θα επιστρέφει *16 channel bits*.



Αρχικά ο κώδικας διαμόρφωσης καναλιού θα λάβει την 8-bit ποσότητα [dec]19 και θα αναλάβει την διαμορφώσει σε μία αντίστοιχη 16-bit ποσότητα. Η λέξη χρήστη [dec]19, κωδικοποιείται στην Κατάσταση [1] στις κωδικές λέξεις 0x2091,1 και 0x0110, 2, στο κύριο βιβλίο κωδικών και στο εναλλακτικό αντίστοιχα. Η θετική αναπαράσταση NRZI για την πρώτη είναι:

Κωδική Λέξη: 0x2091,1

NRZI[+]: +1+1-1-1 -1-1-1-1 +1+1+1-1 -1-1-1+1

RDS[+]: -4

Αντίστοιχα για τη δεύτερη έχουμε:

Κωδική Λέξη: 0x0110,2

NRZI[+]: +1+1+1+1 +1+1+1-1 -1-1-1+1 +1+1+1+1

RDS[+]: +8

Ως βασικό στόχο, όπως έχουμε ήδη πει πολλές φορές, έχουμε θέσει το αρχείο, που θα προκύψει μέσα από τη διαδικασία της διαμόρφωσης, να έχει την ελάχιστη κατ' απόλυτη δυνατή τιμή RDS. Προφανώς θα ξεκινήσουμε με RDS ίσο με μηδέν. Οπότε έχουμε δύο επιλογές, είτε να διαλέξουμε την κωδική λέξη 0x2091,1, είτε την κωδική λέξη 0x0110,2. Για την πρώτη επιλογή έχω $|0-4|=4$, ενώ για τη δεύτερη $|0+8|=8$. Με βάσει όσα έχουμε πει κρίνεται προτιμητέα η πρώτη επιλογή. Θα επιλεγεί λοιπόν η κωδική λέξη 0x2091,1. Η επόμενη κατάσταση του διαμορφωτή θα καθορισθεί αποκλειστικά από την κατάσταση κωδικής λέξης, η οποία στην περίπτωση μας ισούται με 1. Άρα η Κατάσταση [1] θα είναι η επόμενη μας κατάσταση. Στο υπό διαμόρφωση αρχείο θα γραφτεί η NRZI[+] αναπαράσταση της κωδικής λέξης που επιλέχθηκε, δηλαδή η +1+1-1-1 -1-1-1-1 +1+1+1-1 -1-1-1+1. Το τελευταίο ψηφίο της οποίας ισούται με +1. Αυτό το ψηφίο θα καθορίσει αποκλειστικά το πρόσημο, βάσει του οποίου, θα αναπαρασταθεί σε NRZI μορφή η κωδική λέξη, που θα

επιλεχθεί στο αμέσως επόμενο στάδιο. Η RDS ποσότητα θα λάβει την τιμή $RDS=0 + (-4)=-4$, δηλαδή όχι την κατ' απόλυτη RDS τιμή, αλλά την αντίστοιχη προσημασμένη. Η επόμενη λέξη χρήστη ισούται με [dec]24 και στην Κατάσταση [1] κωδικοποιείται στις κωδικές λέξεις 0x1024,2 και 0x0901,1, στο κύριο βιβλίο κωδικών και στο εναλλακτικό αντίστοιχα. Λόγω του θετικού πρόσημου, που έχουμε από την προηγούμενη κωδική λέξη θα εργαστούμε με τη θετική αναπαράσταση.

Η θετική αναπαράσταση NRZI για την πρώτη είναι:

Κωδική Λέξη: 0x1024,2

NRZI[+]: +1+1+1-1 -1-1-1-1 -1-1+1+1 +1-1-1-1

RDS[+]: -8

Αντίστοιχα για τη δεύτερη έχουμε:

Κωδική Λέξη: 0x0901,1

NRZI[+]: +1+1+1+1 -1-1-1+1 +1+1+1+1 +1+1+1-1

RDS[+]: +8

Το RDS από τη προηγούμενη διαμόρφωση ισούται με -4. Η πρώτη λέξη θα μου δώσει $|-4 + (-8)|=12$, ενώ η δεύτερη $|-4 + 8|=4$. Θα προτιμηθεί προφανώς η δεύτερη επιλογή και το RDS θα λάβει την τιμή $RDS=-4+8=4$. Ως επόμενη κατάσταση θα οριστεί η κατάσταση [1] και η NRZI ακολουθία +1+1+1+1 -1-1-1+1 +1+1+1+1 +1+1+1-1 θα γραφτεί στο υπό διαμόρφωση αρχείο. Το τελικό ψηφίο της τελευταίας ισούται με -1, οπότε η NRZI αναπαράσταση της επόμενης κωδικής λέξης θα προσημανθεί αρνητικά.

Η παραπάνω διαδικασία εκτελείται επαναληπτικά και ολοκληρώνεται όταν έχουν διαμορφωθεί όλες οι λέξεις χρήστη σε κωδικές λέξεις καναλιού.

Εύλογα μπορεί κανείς να αναρωτηθεί το λόγο για τον οποίο στο αρχικό στάδιο χρησιμοποιήθηκε ως αφετηρία η Κατάσταση [1] και το θετικό πρόσημο. Η απάντηση είναι πως πρόκειται καθαρά για μια σύμβαση, η οποία όπως δείξαμε στα πειράματά μας δεν επηρεάζει σε καμία περίπτωση την απόδοση του διαμορφωτή/κωδικοποιητή. Δε θα άλλαζε δηλαδή απολύτως τίποτα αν επιλέγαμε να ξεκινήσουμε από κάποια άλλη κατάσταση ή αν εργαζόμασταν με αρνητικό πρόσημο.

> 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 *EFMplus* (αναφορές στο [3]). 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 η κωδική λέξη X και τα ψηφία της NRZI αναπαράστασής της X15, X14, X13, ... , X01, X00. Αν συμβολίσουμε με RDS' το Τρέχον Ψηφιακό Άθροισμα της κωδικής λέξης για αρχικό RDS=0, τότε το νέο κριτήριο μπορεί να εκφραστεί μαθηματικά ως:

Άθροισμα Τετραγώνων =

$$(X15)^2 + (X14)^2 + \dots + (X00)^2 + 2 \cdot RDS' \cdot (X15 + X14 + \dots + X00) + 16 \cdot (RDS')^2$$

Για να κάνουμε ακόμα καλύτερη πρόβλεψη αντικαθιστούμε τον τελευταίο όρο του αθροίσματος με τον όρο $16 \cdot (NEWRDS')^2$. Δηλαδή αντί της τιμής RDS', υπολογίζουμε και χρησιμοποιούμε τη τιμή $NEWRDS = OLDRDS + RDS'$, δηλαδή το άθροισμα της RDS' με την παλιά RDS τιμή.

Οπότε ο παραπάνω τύπος αναδιαμορφώνεται στον:

Άθροισμα Τετραγώνων =

$$(X15)^2 + (X14)^2 + \dots + (X00)^2 + 2 \cdot RDS' \cdot (X15 + X14 + \dots + X00) + 16 \cdot (NEWRDS')^2$$

Διαδικασία Αποδιαμόρφωσης/Αποκωδικοποίησης

Κατά τη σχεδίαση του βιβλίου κωδικών δώσαμε ιδιαίτερη έμφαση στην ανάθεση των κωδικών λέξεων με τέτοιο τρόπο ώστε να εξασφαλίζεται η ανεξαρτησία της διαδικασίας της αποκωδικοποίησης από τις καταστάσεις της κωδικοποίησης. Στην παρούσα ενότητα θα παρουσιάσουμε δια παραδείγματος την αποκωδικοποίηση μέγιστης πιθανοφάνειας (*maximum likelihood decoding*).

Πριν προχωρήσουμε άλλο τη συζήτησή μας και για λόγους που θα φανούν στην πορεία θα αναφερθούμε στην Απόσταση Hamming (*Hamming Distance*). Έστω οι λέξεις 0x0901 και 0x4111, με δυαδικές αναπαραστάσεις 0000 1001 0000 0001 και 0100 0001 0001 0001 αντίστοιχα. Αν αριθμήσουμε τα ψηφία της λέξης από δεξιά προς αριστερά, με το δεξιότερο ψηφίο να σημειώνεται με 0 και το αριστερότερο με 15, προκύπτει πως οι δύο λέξεις διαφέρουν στα ψηφία των θέσεων 14, 11 και 4. Θα ονομάζουμε απόσταση Hamming το πλήθος των ψηφίων στα οποία δύο λέξεις διαφέρουν. Για την παραπάνω περίπτωση η απόσταση Hamming ισούται με 3, αφού, όπως δείξαμε, οι λέξεις διαφέρουν σε ακριβώς 3 ψηφία.

Έστω λοιπόν το παρακάτω απόσπασμα ενός βιβλίου κωδικών τεσσάρων καταστάσεων.

/* 19 */	0x2404, 1,	0x8209, 2,	0x4201, 2,	0x8209, 2,
/* 20 */	0x1004, 1,	0x1021, 1,	0x4209, 1,	0x2048, 1,
/* 21 */	0x2088, 2,	0x1021, 2,	0x4209, 2,	0x2088, 2,
/* 22 */	0x2208, 3,	0x1041, 2,	0x4211, 2,	0x2208, 3,
/* 23 */	0x1004, 2,	0x1004, 2,	0x4088, 2,	0x4088, 2,

Ας υποθέσουμε τώρα πως ζητούμενο είναι αποκωδικοποίηση ενός κωδικοποιημένου αρχείου του οποίου τα πρώτα 32 δυαδικά ψηφία είναι:

0001 0000 0000 0100 0001 0000 0010 0001

Όπως ήδη είπαμε ο κώδικας διαμόρφωσης έχει σχεδιαστεί με τέτοιο τρόπο, ώστε η διαδικασία της αποκωδικοποίησης να ολοκληρώνεται χωρίς καμιά πληροφορία για τις καταστάσεις. Ισχύει πάντα η σύμβαση πως χρησιμοποιούμε ως αφετηρία την κατάσταση [1]. Για την αποκωδικοποίηση των πρώτων 16 ψηφίων στην κατάλληλη λέξη εισόδου, ο αποκωδικοποιητής θα χρειαστεί τα πρώτα 32 ψηφία του αρχείου. Τα πρώτα 16 ψηφία είναι 0001 0000 0000 0100, που αντιστοιχούν στη δεκαεξαδική λέξη 0x1004. Ο αποκωδικοποιητής θα υπολογίσει την απόσταση Hamming μεταξύ των λέξεων της κατάστασης [1] και της λέξης που διάβασε. Θα υπολογιστούν δηλαδή 512 διαφορετικές αποστάσεις και θα επιλεγεί η μικρότερη (ιδανικά η μηδενική). Στην περίπτωση μας λοιπόν θα έχουμε μηδενική απόσταση για τις λέξεις εισόδου 20 και 23. Η απόφαση για την επιλογή μίας εκ των δύο θα βασιστεί στα επόμενα 16 ψηφία του κωδικοποιημένου αρχείου, που είναι τα 0001 0000 0010 0001 και ισοδυναμούν με τη δεκαεξαδική λέξη 0x1021. Παρατηρούμε πως η κωδική λέξη της λέξης εισόδου 20 έχει ως κατάσταση κωδικής λέξης την κατάσταση [1], ενώ η 23 την [2]. Δεν έχουμε λοιπόν παρά να υπολογίσουμε τις αποστάσεις Hamming για τη λέξη 0x1021 και τις λέξεις των καταστάσεων [1] και [2], κρατώντας τελικά τη μικρότερη. Η λέξη 0x1021 θα δώσει τη μικρότερη απόσταση για την κωδική λέξη 0x1021 της κατάστασης [2]. Θα επιλέξουμε οπότε να αποκωδικοποιήσουμε τη λέξη 0x1004 στη λέξη εισόδου 23, ανανεώνοντας παράλληλα την κατάσταση του αποκωδικοποιητή, η οποία τώρα θα γίνει η [2]. Η διαδικασία θα επαναληφθεί μέχρι να επιτευχθεί η πλήρης αποκωδικοποίηση του αρχείου.

> Ανθεκτικότητα σε Σφάλματα:

Κατά την παραπάνω επίδειξη συνειδητά δώσαμε έμφασή στη σωστή αποκωδικοποίηση ενός αρχείου, χωρίς να κάνουμε καμία αναφορά σε ζητήματα θορύβου. Για την πιο ρεαλιστική προσομοίωση και αξιολόγηση ενός κώδικα διαμόρφωσης, προσθέτουμε ανεξάρτητο τυχαίο θόρυβο στο προς αποκωδικοποίηση NRZI αρχείο με πιθανότητα σφάλματος, δηλαδή πιθανότητα να αλλάξει ένα ψηφίο, ίση με ϵ .

Αν συμβολίσουμε με $W1$ το πλήθος των Εσφαλμένα Αποκωδικοποιημένων Λέξεων Εισόδου και με $W2$ το πλήθος των Αλλοιωμένων Κωδικών Λέξεων, η Ανθεκτικότητα σε Σφάλματα (*Error Resilience*) μπορεί να οριστεί ως:

$$\text{Error Resilience} = W1/W2$$

Ο παραπάνω λόγος ουσιαστικά μας δείχνει πόσες από τις λέξεις εισόδου, αποκωδικοποιήθηκαν λάθος από τον αποκωδικοποιητή, λόγω των αλλοιωμένων κωδικών λέξεων.

> Αποτελέσματα Ανθεκτικότητας σε Σφάλματα:

Από την αρχή κιόλας της εργασίας αναφερθήκαμε στην υπεροχή του βιβλίου κωδικών που παράχθηκε με την εφαρμογή της μεθόδου διαχωρισμού [2]. Μετρήσαμε και παρουσιάζουμε παρακάτω την ανθεκτικότητα σε σφάλματα που αυτό επιτυγχάνει:

Μέθοδος Διαχωρισμού	Πιθανότητα Σφάλματος	Error Resilience
[2]	$10^{(-2)}$	0.5988
[2]	$10^{(-3)}$	0.5824
[2]	$10^{(-4)}$	0.5843

Θεωρητικός Υπολογισμός RDS Διασποράς

Πριν αναφερθούμε αναλυτικά στον θεωρητικό υπολογισμό της RDS Διασποράς και για λόγους που θα γίνουν προφανείς στη συνέχεια της εργασίας θα παρουσιάσουμε εκτενώς την Αλυσίδα Μαρκόβ Διακριτού Χρόνου (*Discrete Time Markov Chain or DTMC*).

> Στοχαστικές Διαδικασίες:

Ορισμός: Μια **στοχαστική διαδικασία** είναι μία μόνο-παραμετρική οικογένεια $\{X(t)\}$ τυχαίων μεταβλητών, που είναι όλες ορισμένες στον ίδιο χώρο πιθανότητας (Ω, F, P) . Η παράμετρος t συνήθως παριστάνει το χρόνο και παίρνει είτε διακριτές ή συνεχείς τιμές (θα δούμε αμέσως μετά τι σημαίνει αυτό). Με άλλα λόγια, για κάθε τιμή της παραμέτρου t , $X(t) : \Omega \rightarrow R$ με $X(t)^{-1}(B) \in F$, για κάθε σύνολο Borel $B \in B(R)$.

Ενδιαφερόμαστε για την περίπτωση κατά την οποία η παράμετρος t παίρνει διακριτές τιμές, τότε συνήθως γράφουμε (σαν δέκτη) n αντί για t (σε παρενθέσεις), όπου θεωρούμε ότι $n = 0, 1, \dots$, και λέμε ότι η στοχαστική διαδικασία $\{X_n\}_{n \geq 0}$ αποτελεί μία **στοχαστική διαδικασία διακριτού χρόνου**.

> Μαρκοβιανές Στοχαστικές Διαδικασίες:

Οι **μαρκοβιανές στοχαστικές διαδικασίες** ή **στοχαστικές διαδικασίες Μαρκόβ**, οφείλουν το όνομα τους στο ρώσο μαθηματικό **Αντρέα Μαρκόβ (Andrei Andreyevich Markov, 1856 - 1922)**. Στη βιβλιογραφία πολλές φορές περιγράφονται με τον όρο **μαρκοβιανή αλυσίδα (markov chain)**. Σε γενικές γραμμές δύο είναι τα συγκεκριμένα ειδικά χαρακτηριστικά των διαδικασιών Μαρκόβ:

(i) Το γεγονός ότι οι τυχαίες μεταβλητές $\{X_n\}_{n \geq 0}$ ή $\{X(t)\}_{t \geq 0}$ είναι διακριτές και παίρνουν τιμές σε ένα αριθμήσιμο (πεπερασμένο ή άπειρο) σύνολο S , το οποίο ονομάζεται **χώρος καταστάσεων** και τα στοιχεία του ονομάζονται **καταστάσεις** της διαδικασίας. **Επιπλέον, αν $X_n = s_j$ (ή $X(t) = s_j$), για κάποιο $s_j \in S$, λέμε ότι η τυχαία μεταβλητή X_n (ή $X(t)$) βρίσκεται στην κατάσταση s_j στο χρόνο n (ή t).**

(ii) Και το γεγονός, που ονομάζεται ιδιότητα Μαρκόβ, ότι η στοχαστική διαδικασία στερείται μνήμης με την έννοια που θα διευκρινιστεί καλύτερα στους ορισμούς που ακολουθούν.

Οι μαρκοβιανές αλυσίδες διακρίνονται σε μαρκοβιανές αλυσίδες διακριτού και σε μαρκοβιανές αλυσίδες συνεχούς χρόνου. Στην παρούσα εργασία ενδιαφερόμαστε για τις αντίστοιχες διακριτού χρόνου.

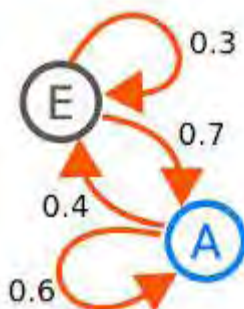
> Μαρκοβιανή Αλυσίδα Διακριτού Χρόνου:

Ορισμός: Μια στοχαστική διαδικασία διακριτού χρόνου $\{X_n\}_{n \geq 0}$ λέγεται ότι είναι μια **μαρκοβιανή αλυσίδα**, αν ικανοποιούνται οι εξής δύο συνθήκες:

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

(ii) Για κάθε $n = 0, 1, \dots$, για κάθε $s_i, s_{i0}, s_{i1}, \dots, s_{i(n-1)}, s_j \in S$, ισχύει η παρακάτω ιδιότητα Μαρκόβ:

$$P(X_{n+1} = s_j | X_n = s_i, X_{n-1} = s_{i(n-1)}, \dots, X_1 = s_{i(1)}, X_0 = s_{i(0)}) = P(X_{n+1} = s_j | X_n = s_i)$$



Απλή Μαρκοβιανή Αλυσίδα Δύο Καταστάσεων

> Στάσιμες Κατανομές:

Έστω πάντα η **αλυσίδα Markov** $\{X_n\}_{n \geq 0}$ στον αριθμησιμο χώρο καταστάσεων $S = \{1, 2, \dots\}$ με πίνακα πιθανοτήτων μεταβάσεων P .

Ορισμός: Ένα διάνυσμα σειράς $\pi = (\pi_j; j \in S)$ λέγεται ότι αποτελεί μια **στάσιμη κατανομή (stationary distribution)** της αλυσίδας Markov, αν το π ικανοποιεί τις εξής σχέσεις:

(i) $\pi_j \geq 0$, για κάθε $j \in S$, και $\sum_{(j \in S)} \pi_j = 1$, και

(ii) $\pi P = \pi$, με την έννοια ότι ισχύει η εξής **εξίσωση ολικού ισοζυγίου**

$$\sum_{(i \in S)} \pi_i P_{ij} = \pi_j, \text{ για κάθε } j \in S$$

> Μέγιστο RDS:

Στην αρχή της εργασίας δείξαμε τον τρόπο υπολογισμού της RDS ποσότητας. Θυμίζουμε την εν λόγω μεθοδολογία μέσα από το παρακάτω παράδειγμα:

Έστω η λέξη 0x8922, με δυαδική αναπαράσταση 1000100100100010. Η θετικά προσημασμένη NRZI αναπαράσταση είναι η -1-1-1-1+1+1+1-1-1-1+1+1+1+1-1-1 και η αντίστοιχη αρνητική +1+1+1+1-1-1-1+1+1+1-1-1-1-1+1+1.

Αν αθροίσουμε όλα τα ψηφία (σύνολο 16 ψηφίων) της NRZI αναπαράστασης τότε έχουμε: $RDS[+] = -2$ και $RDS[-] = +2$.

Εύκολα μπορεί κανείς να δείξει ότι για οποιαδήποτε έγκυρη λέξη, η ποσότητα του RDS λαμβάνει αυστηρά άρτιες θετικές ή αρνητικές διακριτές τιμές, με μέγιστη θετική τιμή +10 και ελάχιστη αρνητική -10.

Ως έγκυρη λέξη ορίζουμε τη δυαδική λέξη 16 ψηφίων, που υπακούει στον περιορισμό $(d, k)=(2, 10)$.

> Ορισμός Χώρου Καταστάσεων:

Κατά την παρουσίαση της διαδικασίας κωδικοποίησης/διαμόρφωσης σημειώσαμε ρητά την άμεση εξάρτηση της απόφασης που λαμβάνει ο κωδικοποιητής από την τριάδα Τρέχουσα Τιμή RDS(1), Επόμενη Κατάσταση(2) και Πολικότητα(3).

Στην προσπάθειά μας να ορίσουμε το χώρο καταστάσεων προχωρήσαμε στην απαισιόδοξη εκτίμηση πως η ποσότητα του RDS για το βιβλίο κωδικών που προσομοιώνουμε θα λάβει μέγιστη θετική τιμή +100 και ελάχιστη αρνητική τιμή -100. Το πλήθος όλων των δυνατών τιμών, λόγω και της αρτιότητας της ποσότητας του RDS, ισούται με 101. Το υπό προσομοίωση βιβλίο κωδικών βασίζεται σε μια πεπερασμένη μηχανή 15 καταστάσεων και η πολικότητα περιορίζεται σε δύο πιθανά ενδεχόμενα, στον ενδεχόμενο να είναι θετική και στο αντίστοιχο να είναι αρνητική. Οπότε ο χώρος καταστάσεων του βιβλίου θα αποτελείται από $101 \times 2 \times 15 = 3030$ καταστάσεις, με κάθε μία κατάσταση να προσδιορίζεται μοναδικά από την τριάδα που περιγράψαμε στην αρχή. Για να αποφύγουμε πιθανή σύγχυση των καταστάσεων του διαμορφωτή/κωδικοποιητή με τις καταστάσεις του χώρου καταστάσεων τούδε και στο εξής θα αναφερόμαστε στις δεύτερες, με τον όρο υπέρ-καταστάσεις.

> Υπολογισμός Πίνακα Πιθανοτήτων Μεταβάσεων:

Ο Πίνακας Πιθανοτήτων Μεταβάσεων θα είναι κατά τα γνωστά τετραγωνικός με τις διαστάσεις του να καθορίζονται αποκλειστικά από το πλήθος υπέρ-καταστάσεων. Οπότε θα έχουμε έναν 3030×3030 πίνακα, με όλα τα στοιχεία του αρχικά να ισούνται με μηδέν. Ο υπολογισμός του εν λόγω πίνακα θα επιδειχθεί αναλυτικά παρακάτω.

Ας υποθέσουμε λοιπόν τώρα πως δίνεται τιμή RDS R, κατάσταση K και πολικότητα Σ, με τη σχετική τριάδα(R,K,Σ) να αντιστοιχείται μοναδικά στην υπέρ-κατάσταση Y1. Αν υποθέσουμε πως ζητούμενο είναι η διαμόρφωση/κωδικοποίηση μιας λέξης εισόδου A, μπορούμε να υπολογίσουμε εύκολα τη νέα υπέρ-κατάσταση στην οποία θα οδηγηθούμε. Ας τη συμβολίσουμε με Y2. Σημειώνουμε πως η διαμόρφωση/κωδικοποίηση της λέξης εισόδου πρέπει να γίνεται ακολουθώντας επ' ακριβώς τη σχετική διαδικασία επιλογής κύριας ή εναλλακτικής κωδικής λέξης, που παρουσιάστηκε παραπάνω. Βασίζουμε την προσομοίωση μας στην ομοιόμορφη κατανομή των λέξεων εισόδου και ολοκληρώνουμε τον υπολογισμό μας προσθέτοντας τον όρο 1.0/256 στη θέση [Y1][Y2] του πίνακα πιθανοτήτων μεταβάσεων. Στη συνέχεια επαναλαμβάνουμε τη διαδικασία για τις εναπομείναντες 255 λέξεις εισόδου. Ο υπολογισμός θεωρείται πλήρης όταν έχουμε εξετάσει όλες τις υπέρ-καταστάσεις(πλήθος υπέρ-καταστάσεων ίσο με 3030) του βιβλίου κωδικών.

Από τον Πίνακα Πιθανοτήτων Μετάβασης μπορούμε εύκολα να υπολογίσουμε τη Στάσιμη Κατανομή. Ουσιαστικά αναζητούμε εκείνο το διάνυσμα π που θα ικανοποιεί την εξίσωση $\pi P = \pi$, όπου P ο πίνακας πιθανοτήτων μετάβασης. Μας ενδιαφέρει το διάνυσμα π να είναι μη μηδενικό ($\pi \neq 0$), με τη μοναδικότητά του να είναι εγγυημένη για κανονική (regular) μορφή του πίνακα P (αναφορές στο [14]). Πιο συγκεκριμένα θέλουμε κάθε στοιχείο του P να είναι μεγαλύτερο του μηδενός και μικρότερο της μονάδας ($0 < p_{ij} < 1$), κάτι που προφανώς ισχύει στην περίπτωση του πίνακα πιθανοτήτων.

Το διάνυσμα π (στάσιμη κατανομή) θα είναι διαστάσεων 1×3030 , αφού όπως ήδη είπαμε πολλές φορές έχουμε 3030 υπέρ-καταστάσεις. Το διάνυσμα π ουσιαστικά περιέχει την πιθανότητα κάθε υπέρ-κατάστασης. Εύκολα μπορεί κανείς να δείξει ότι μία δεδομένη τιμή RDS R αντιστοιχεί σε 30 διαφορετικές υπέρ-καταστάσεις (πλήθος καταστάσεων (=15) \times πλήθος πολικότητας (=2)). Ενδιαφερόμαστε για αυτήν ακριβώς τη πληροφορία, δηλαδή για τη πιθανότητα κάθε τιμής RDS (πλήθος διαφορετικών RDS τιμών (=101)). Υπολογίζουμε λοιπόν το νέο διάνυσμα διαστάσεων 1×101 , της οριακής κατανομής του διανύσματος π .

> Υπολογισμός RDS Διασποράς:

Έχοντας υπολογίσει το διάνυσμα πιθανοτήτων RDS (διάνυσμα διαστάσεων 1×101) μπορούμε εύκολα να υπολογίσουμε το μέσο όρο (mean) και τη διασπορά (variance) της RDS ποσότητας. Αν πολλαπλασιάσουμε κάθε τιμή RDS με την πιθανότητά της, θα οδηγηθούμε σε 101 διαφορετικά ενδιάμεσα αποτελέσματα/γινόμενα. Το άθροισμα αυτών των ενδιάμεσων αποτελεσμάτων/γινόμενων θα ισούται με το μέσο όρο RDS. Αναμένουμε ο μέσος όρος να περίπου ίσος με μηδέν, με τη διασπορά να υπολογίζεται με απλή εφαρμογή του σχετικού τύπου. Παρακάτω φαίνονται οι μαθηματικές διατυπώσεις όσων αναφέραμε:

*Συμβολίζουμε με R το διάνυσμα πιθανότητας RDS και με RDS κάθε τιμή RDS.

$$\text{Μέσος Όρος} = \mu = \sum(RDS * R_{RDS}), \text{ για όλες τις τιμές RDS(101 τιμές)}$$

$$\text{Διασπορά} = \text{Var} = \sum(RDS * RDS * R_{RDS}), \text{ για όλες τις τιμές RDS(101 τιμές)}$$

*Στον τύπο της διασποράς λαμβάνεται υπόψη η μηδενικότητα του μέσου όρου.

> Αποτελέσματα RDS Διασποράς:

<u>Μέθοδος Διαχωρισμού</u>	<u>RDS Διασπορά</u>
[1]	56.3
[2]	46.2
[3]	48.2

Βελτίωση Κώδικα Διαμόρφωσης

Θεωρήσαμε πως η ανθεκτικότητά σε σφάλματα (error resilience) που επιτυγχάνει ο νέος κώδικας διαμόρφωσης είναι ικανοποιητική και εστιάσαμε στη βελτίωση του βιβλίου κωδικών στην κατεύθυνση της επίτευξης της ελάχιστης δυνατής RDS Διασποράς. Στην αυτήν μας την προσπάθειά χρησιμοποιήσαμε ως βάση το εργαλείο της θεωρητικής προσομοίωσης που παρουσιάστηκε εκτενώς παραπάνω.

Η μέθοδος βελτίωσης που εφαρμόσαμε βασίζεται σε ιδέα του Ιωάννη Κατσαβουνίδη. Η εν λόγω μέθοδος εφαρμόστηκε αρχικά, σε συνεργασία με τη Μάρθα Πέππα, για τη βελτίωση του βιβλίου κωδικών του EFMplus(αναφορές στο [1]) δίνοντας εξαιρετικά αποτελέσματα. Ο κώδικας διαμόρφωσης EFMplus πριν την εφαρμογή της μεθόδου επιτύχανε RDS Διασπορά περίπου ίση με 19, με την εφαρμογή της μεθόδου να μειώνει αυτό το νούμερο κατά περίπου 10 μονάδες. Μας οδήγησε δηλαδή σε έναν κώδικα διαμόρφωσης με RDS Διασπορά λίγο πάνω από 9.

Παρουσιάζουμε τη μέθοδο βελτίωσης μέσα από την εφαρμογή της σε ένα υποθετικό βιβλίο κωδικών τεσσάρων καταστάσεων. Έστω λοιπόν το μικρό απόσπασμα του εν λόγω βιβλίου κωδικών:

```
...
/* 19 */ 0x2091, 1, 0x2091, 1, 0x8880, 4, 0x8880, 4,
/* 20 */ 0x2121, 1, 0x2121, 1, 0x9100, 4, 0x9100, 4,
/* 21 */ 0x2241, 1, 0x2241, 1, 0x4049, 1, 0x4049, 1,
/* 22 */ 0x2481, 1, 0x2481, 1, 0x8021, 1, 0x8021, 1,
/* 23 */ 0x1004, 2, 0x1004, 2, 0x4088, 2, 0x4088, 2,
/* 24 */ 0x1024, 2, 0x1024, 2, 0x4088, 3, 0x4088, 3,
/* 25 */ 0x1048, 2, 0x4120, 2, 0x1048, 2, 0x4120, 2,
/* 26 */ 0x1090, 2, 0x1090, 2, 0x8441, 1, 0x8441, 1,
/* 27 */ 0x1120, 2, 0x1120, 2, 0x8482, 1, 0x8482, 1,
/* 28 */ 0x2008, 2, 0x4090, 3, 0x2008, 2, 0x4090, 3,
```

Ας υποθέσουμε λοιπόν τώρα πως η ποσότητα της RDS Διασποράς για το παραπάνω βιβλίο κωδικών μετρήθηκε σε μία τιμή Χ. Έστω τώρα ότι αντικαθιστούμε τις κωδικές λέξεις που κωδικοποιούν τη λέξη εισόδου 28, με τις αντίστοιχες που κωδικοποιούν τη λέξη εισόδου 24. Πρακτικά δηλαδή, εναλλάσσουμε το περιεχόμενο της γραμμής 28 με αυτό της γραμμής 24. Οπότε έχουμε:

```
...
/* 19 */ 0x2091, 1, 0x2091, 1, 0x8880, 4, 0x8880, 4,
/* 20 */ 0x2121, 1, 0x2121, 1, 0x9100, 4, 0x9100, 4,
/* 21 */ 0x2241, 1, 0x2241, 1, 0x4049, 1, 0x4049, 1,
/* 22 */ 0x2481, 1, 0x2481, 1, 0x8021, 1, 0x8021, 1,
/* 23 */ 0x1004, 2, 0x1004, 2, 0x4088, 2, 0x4088, 2,
/* 24 */ 0x2008, 2, 0x4090, 3, 0x2008, 2, 0x4090, 3,
/* 25 */ 0x1048, 2, 0x4120, 2, 0x1048, 2, 0x4120, 2,
/* 26 */ 0x1090, 2, 0x1090, 2, 0x8441, 1, 0x8441, 1,
/* 27 */ 0x1120, 2, 0x1120, 2, 0x8482, 1, 0x8482, 1,
/* 28 */ 0x1024, 2, 0x1024, 2, 0x4088, 3, 0x4088, 3,
```

Εύκολά μπορεί κανείς να πειστεί πως η παραπάνω αλλαγή στο βιβλίο κωδικών θα επηρεάσει την RDS Διασπορά. Το εάν ωστόσο θα επιφέρει αύξηση ή μείωση σε αυτήν δεν είναι γνωστό εκ των προτέρων.

Η μετάθεση της γραμμής 24 με τη γραμμή 28 μετέβαλε την RDS Διασπορά. Προφανώς μας επιτρέπεται να κάνουμε το ίδιο και για τις υπόλοιπες γραμμές του πίνακα. Το βιβλίο κωδικών απαρτίζεται από 512 γραμμές, μπορούμε οπότε να

δοκιμάσουμε $261632 (= 512 \times 511)$ μεταθέσεις. Θα μπορούσαμε ακόμη να επαναλάβουμε το παραπάνω, αλλά αυτή τη φορά να μεταθέτουμε μόνον εκείνες τις κωδικές λέξεις που ανήκουν στις καταστάσεις 1, 2 και 3 ή μόνον εκείνες που ανήκουν στις καταστάσεις 1 και 4 ή μόνον εκείνες των καταστάσεων 2, 3 και 4, και ούτω καθ' εξής.

Για την καλύτερη οργάνωση όλων των δυνατών τρόπων μεταθέσεων θα εισάγουμε την έννοια της *στρατηγικής*. Ο ορισμός της εν λόγω έννοιας δίνεται δια των παρακάτω παραδειγμάτων. Η στρατηγική 15 παραδείγματος χάριν, με δυαδική αναπαράσταση 1111, θα συμβολίζει τη μετάθεση των κωδικών λέξεων που ανήκουν στις καταστάσεις 1, 2, 3 και 4. Κατ' αντίστοιχο τρόπο η στρατηγική 3(=[bin]0011) θα συμβολίζει τη μετάθεση κωδικών λέξεων, που ανήκουν στις καταστάσεις 1 και 2, ενώ η στρατηγική 11(=[bin]1011) τη μετάθεση μεταξύ των καταστάσεων 1, 3 και 4. Προφανώς λοιπόν στο βιβλίο κωδικών τεσσάρων καταστάσεων μπορούμε να δοκιμάσουμε $15(2^4 - 1)$ διαφορετικές στρατηγικές.

Παραπάνω επιλέξαμε να παρουσιάσουμε τη βασική ιδέα της μεθόδου βελτίωσης του βιβλίου κωδικών, αποκρύπτοντας συνειδητά τους περιορισμούς που θέτει η συνεπής ανάθεση στην εφαρμογή ή μη μιας στρατηγικής. Για να γίνουν πιο σαφείς οι εν λόγω περιορισμοί δίνουμε το παρακάτω παράδειγμα. Ας υποθέσουμε για χάρη ευκολίας της συζήτησης πως αυτή τη φορά διαθέτουμε ένα βιβλίο κωδικών 7 καταστάσεων. Δηλαδή έχουμε στη διάθεση μας 127 δυνατές στρατηγικές. Θα εξετάσουμε κατά πόσον οι τυχαία επιλεγμένες στρατηγικές 46(=[bin] 0101110) και 100(=[bin]1100100) βρίσκουν εφαρμογή. Η δυαδική αναπαράσταση του αριθμού της στρατηγικής μας βοηθάει στην εξαγωγή των καταστάσεων στις οποίες ανήκουν οι επιτρεπόμενες προς μετακίνηση κωδικές λέξεις. Ως παράδειγμα η στρατηγική 46, με δυαδική αναπαράσταση 0101110, μας σημειώνει πως οι κωδικές λέξεις που επιτρέπονται να μετακινηθούν πρέπει να ανήκουν αυστηρά στις καταστάσεις 2, 4, 5 και 6. Θα δείξουμε την ανάλυση μας για την τυχαία γραμμή του βιβλίου κωδικών που φαίνεται παρακάτω. Έστω, λοιπόν, η γραμμή:

0x0111, 1, 0x0111, 1, 0xffff, 0, 0x8440, 12, 0xffff, 0, 0x8440, 12, 0x8440, 12

[Στρατηγική 46]: Η στρατηγική 46 επιτάσσει την αυστηρή μετακίνηση μόνο των λέξεων εκείνων που ανήκουν στις καταστάσεις 2, 4, 5 και 6. Δηλαδή των λέξεων που στην παρακάτω γραμμή έχουν χρωματιστεί με μαύρο.

0x0111, 1, 0x0111, 1, 0xffff, 0, 0x8440, 12, 0xffff, 0, 0x8440, 12, 0x8440, 12

Λόγω της συνεπούς ανάθεσης η μετακίνηση της κωδικής λέξης [0x0111, 1] της κατάστασης 2 δεν μπορεί να γίνει χωρίς την ταυτόχρονη μετακίνηση της λέξης [0x0111, 1] της κατάστασης 1. Αντίστοιχα οι κωδικές λέξεις [0x8440, 12] των καταστάσεων 4 και 6 δεν μπορούν να μετακινηθούν χωρίς την μετακίνηση της λέξης [0x8440, 12] της κατάστασης 7. Ωστόσο, η μετακίνηση λέξεων που επιτρέπει η στρατηγική περιορίζεται αυστηρά στις καταστάσεις 2, 4, 5 και 6. Οπότε η στρατηγική χαρακτηρίζεται ως άκυρη για την εν λόγω γραμμή και δεν μπορεί να

εφαρμοστεί. Γραφικά ο παραπάνω συλλογισμός φαίνεται στο παρακάτω σχήμα, όπου με κίτρινο χρωματίζονται οι λέξεις από τις οποίες υπάρχει εξάρτηση.



0x0111, 1, 0x0111, 1, 0xffff, 0, 0x8440, 12, 0xffff, 0, 0x8440, 12, 0x8440, 12

[Στρατηγική 100]: Η στρατηγική 100 επιτάσσει την αυστηρή μετακίνηση μόνο των λέξεων εκείνων που ανήκουν στις καταστάσεις 1, 2 και 5. Δηλαδή των λέξεων που παρακάτω έχουν χρωματιστεί με μαύρο.

0x0111, 1, 0x0111, 1, 0xffff, 0, 0x8440, 12, 0xffff, 0, 0x8440, 12, 0x8440, 12

Σε αυτήν την περίπτωση η συνεπής ανάθεση δεν θέτει κάποιον απαγορευτικό περιορισμό για τη μετακίνηση λέξεων των καταστάσεων 1, 2 και 5. Οπότε η στρατηγική χαρακτηρίζεται ως έγκυρη για την εν λόγω γραμμή και δύναται να εφαρμοστεί.



0x0111, 1, 0x0111, 1, 0xffff, 0, 0x8440, 12, 0xffff, 0, 0x8440, 12, 0x8440, 12

Το βιβλίο κωδικών έχει 256 λέξεις εισόδου(πρακτικά 512 γραμμές, λόγω της ύπαρξης κύριων και εναλλακτικών κωδικών λέξεων), οπότε για να χαρακτηριστεί μία στρατηγική έγκυρη αρκεί να βρεθούν τουλάχιστον δύο λέξεις εισόδου(πρακτικά γραμμές) που να βρίσκει εφαρμογή η στρατηγική. Ο χαρακτηρισμός μιας στρατηγικής ως έγκυρης σημαίνει πως η στρατηγική δύναται υπό προϋποθέσεις να βελτιώσει το βιβλίο κωδικών. Θα χρησιμοποιήσουμε και πάλι ένα παράδειγμα για την καλύτερη κατανόηση της μεθόδου βελτίωσης, επιδεικνύοντάς την αυτήν τη φορά στο βιβλίο κωδικών 15 καταστάσεων. Με βάσει όσα είπαμε είναι πλέον προφανές πως θεωρητικά μπορούμε να εφαρμόσουμε 32767 διαφορετικές στρατηγικές($2^{15} - 1$). Από τις 32767 δυνατές στρατηγικές, μόνο οι 7644 βρίσκουν εφαρμογή στο δικό μας βιβλίο κωδικών. Οι υπόλοιπες 25123 ακυρώνονται από τους περιορισμούς της συνεπούς ανάθεσης. Επιλέγουμε τυχαία προς επίδειξη την εφαρμογή της στρατηγικής 8771(=[bin]010001001000011). Αρχικά έχουμε το βιβλίο κωδικών έτσι όπως φαίνεται στην παρακάτω φιγούρα.

/* 0 */ 0x2009, 1, 0x0409, 1, . . . 0x4009, 1, 0x2009, 1, 0x0409, 1,

/* 1 */ 0x2011, 1, 0x0409, 2, . . . 0x4009, 2, 0x2011, 1, 0x0409, 2,

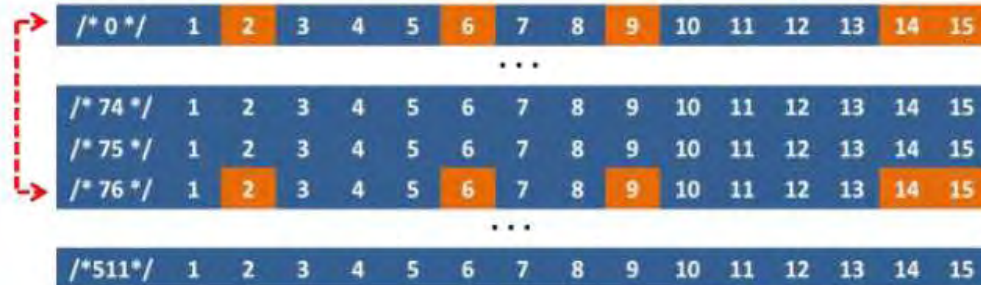
/* 2 */ 0x2021, 1, 0x0411, 1, . . . 0x4011, 1, 0x2021, 1, 0x0411, 1,

. . .

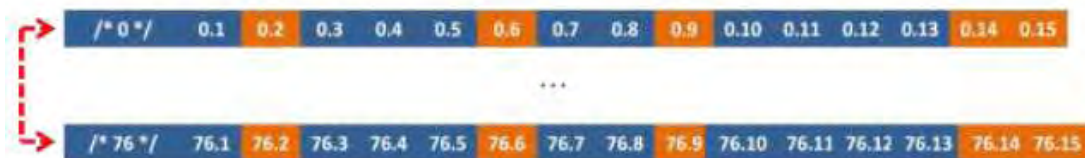
/* 510 */ 0xffff, 0, 0x0111, 1, . . . 0x0111, 1, 0xffff, 0, 0xffff, 0,

/* 511 */ 0xffff, 0, 0x0109, 1, . . . 0x0109, 1, 0xffff, 0, 0xffff, 0,

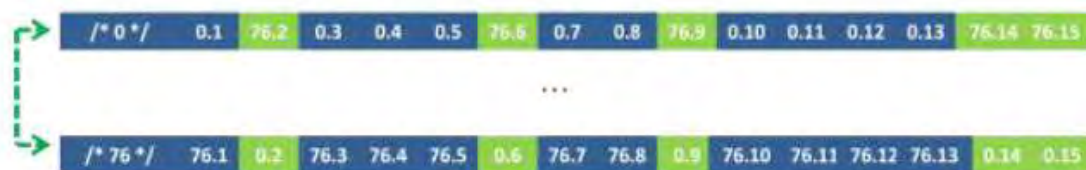
Η δυαδική αναπαράσταση της στρατηγικής με αριθμό 8771 είναι η ακολουθία [010001001000011]. Οι προς μετακίνηση κωδικές λέξεις ανήκουν στις καταστάσεις 2, 6, 9, 14 και 15(στο σχήμα παρακάτω σημειώνονται με πορτοκαλί χρώμα). Παρακάτω θα δείξουμε την εφαρμογή της στρατηγικής για τις γραμμές /* 0 */ και /* 76 */.



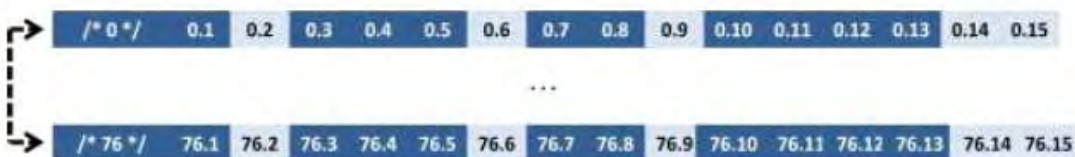
Για να αποσαφηνιστούν περισσότερο τα πράγματα συμβολίζουμε τη λέξη της κατάστασης 5 που ανήκει στη γραμμή 0 ως 0.5. Κατά την εκτέλεση λοιπόν της στρατηγικής θα σημειωθούν αρχικά οι προς μετακίνηση λέξεις(χρωματισμένες με πορτοκαλί) και θα υπολογισθεί η διακύμανση του Τρέχοντος Ψηφιακού Αθροίσματος(RDS Variance), ας την ονομάσουμε διακύμανση(1).



Η στρατηγική στη συνέχεια θα τοποθετήσει στις καταστάσεις 2, 6, 9, 14 και 15 της γραμμής /* 0 */, τις αντίστοιχες λέξεις της γραμμής /* 76 */. Αντίστοιχα στη γραμμή /* 76 */ θα τοποθετηθούν οι λέξεις των καταστάσεων 2, 6, 9, 14 και 15 της γραμμής /* 0 */. Κατόπιν θα υπολογισθεί η διακύμανση του Τρέχοντος Ψηφιακού Αθροίσματος(RDS Variance), την οποία αυτή τη φορά θα ονομάσουμε διακύμανση(2).



Εάν η διακύμανση(2) είναι μικρότερη της διακύμανσης(1), τότε η αλλαγή που επέφερε η στρατηγική θα ενσωματωθεί στο νέο βελτιωμένο βιβλίο κωδικών. Σε διαφορετική περίπτωση η αλλαγή θα αγνοηθεί και θα επανέλθει το βιβλίο κωδικών στην αρχική του μορφή.



Η παραπάνω διαδικασία που περιγράφηκε παραπάνω εκτελείται επαναληπτικά για όλες τις έγκυρες στρατηγικές και σταματά μόνο όταν για έναν ολοκληρωμένο γύρο

όλων των στρατηγικών δε σημειωθεί καμία περαιτέρω βελτίωση του βιβλίου κωδικών.

> Αποτελέσματα Βελτίωσης:

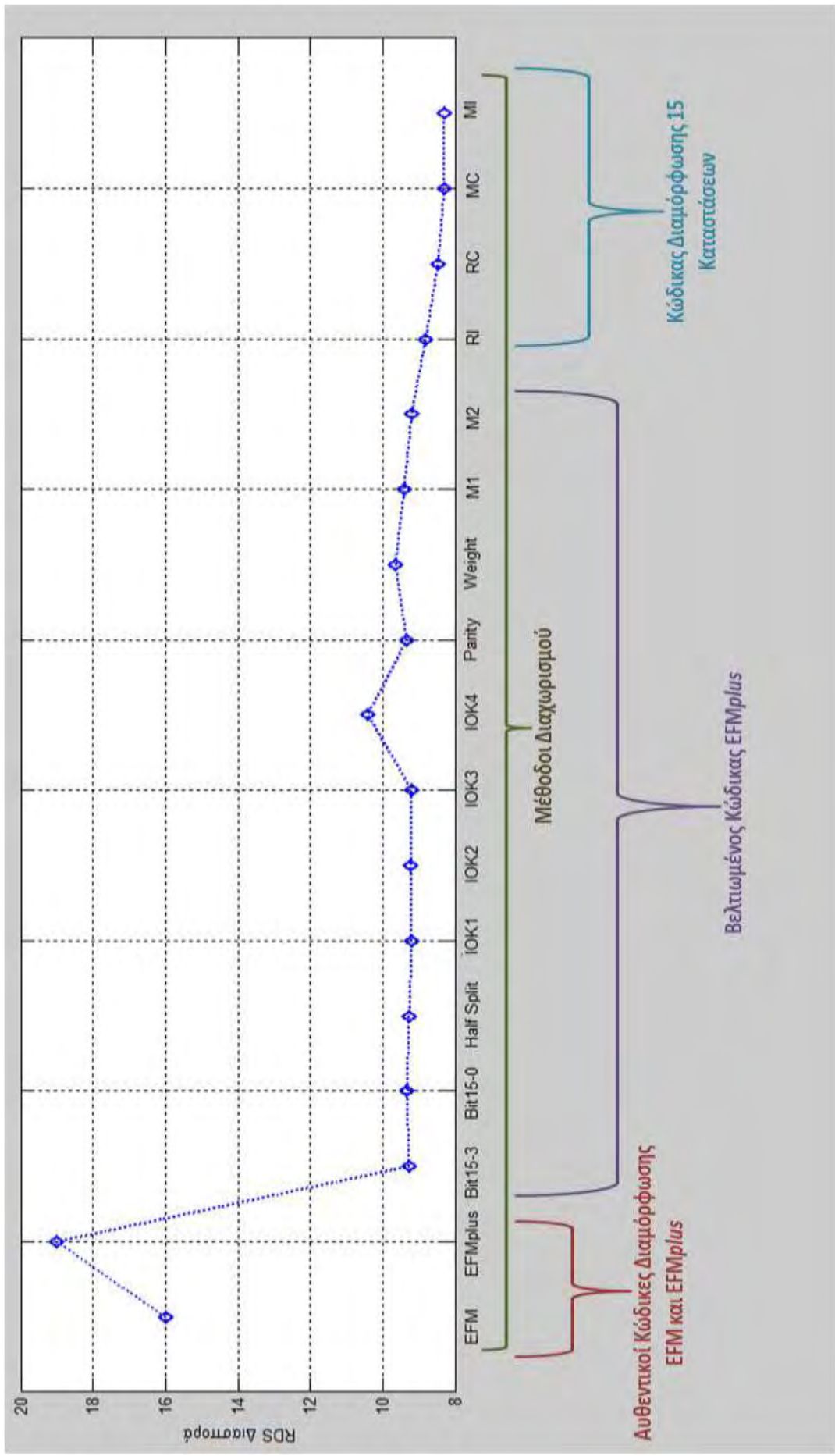
<u>Μέθοδος Διαχωρισμού</u>	<u>RDS Διασπορά</u>
[1]	8.48
[2]	8.29
[3]	8.31

> Συνολικά Αποτελέσματα Βελτίωσης:

Όπως ήδη αναφέραμε πριν την εφαρμογή της παραπάνω μεθόδου βελτίωσης στον κώδικα διαμόρφωσης 15 καταστάσεων, η μέθοδος είχε εφαρμοστεί επιτυχημένα στον κώδικα διαμόρφωσης EFMplus. Θεωρήσαμε σκόπιμο να απεικονίσουμε τη συνολική απόδοση της μεθόδου, στο γράφημα που φαίνεται στην επόμενη σελίδα(σελ. 33). Στο γράφημα περιλαμβάνονται οι αυθεντικοί κώδικες EFM και EFMplus, οι βελτιώσεις που έγιναν στο δεύτερο στα πλαίσια της διπλωματικής εργασίας της Μάρθας Πέππα(αναφορές στο [1]) και τέλος τα αποτελέσματα του κώδικα διαμόρφωσης 15 καταστάσεων.

> Χρόνος Εκτέλεσης Μεθόδου Βελτίωσης:

Η εκτέλεση ενός πλήρους γύρου στρατηγικών για ένα βιβλίο κωδικών 15 καταστάσεων, σε ένα σύστημα Intel(R) Core(TM) i3-2310M CPU @ 2.10GHZ, με εγκατεστημένη RAM 3,00 GB, απαιτεί κάτι λιγότερο από 30 λεπτά. Η εφαρμογή της μεθόδου βελτίωσης σε ένα δείγμα με περισσότερα από τριάντα βιβλία κωδικών έδειξε πως η μέθοδος ολοκληρώνεται μετά από περίπου 6-8 ολοκληρωμένους γύρους.



Επίλογος/Αποτελέσματα Εργασίας

Στην παρούσα διπλωματική εργασία μελετήσαμε τους κώδικες διαμόρφωσης καναλιού EFM και EFMplus. Εστίασαμε τη μελέτη μας στον EFMplus, ο οποίος βρίσκει εφαρμογή στον οπτικό δίσκο DVD. Ο εν λόγω κώδικας βασίζει την λειτουργία του σε ένα διαμορφωτή/κωδικοποιητή τεσσάρων καταστάσεων. Θέσαμε εξ' αρχής ως στόχο της εργασίας το δύσκολο έργο της κατασκευής ενός καινοτόμου κώδικα διαμόρφωσης, στα πρότυπα του EFMplus, ο οποίος θα χρησιμοποιεί περισσότερες των τεσσάρων καταστάσεων και θα επιτυγχάνει αυξημένη απόδοση συγκριτικά με τον πρώτο. Επιτύχαμε το στόχο μας κατασκευάζοντας έναν νέο κώδικα δεκαπέντε καταστάσεων, ο οποίος πράγματι επιτυγχάνει σημαντικά καλύτερη απόδοση. Στο παραπάνω κείμενο παρουσιάσαμε εκτενώς όλα τα θέματα της υλοποίησης του νέου κώδικα, τεκμηριώνοντας παράλληλα την υπεροχή του σε σχέση με τον EFMplus.

Αναφορές

- [1] Μ. Πέππα, Λ. Τασιούλας, Ι. Κατσαβουνίδης, “Μελέτη και υλοποίηση μεθόδων διαμόρφωσης καναλιού με περιορισμούς μήκους”, *Διπλωματική Εργασία*, Βόλος 2012
- [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] A.J. Viterbi, “Convolutional Codes and Their Performance in Communication Systems”, *IEEE Transactions on Communications Technology*, VOL. 19, pp. 751-771, October 1971
- [5] C. Shannon, “A Mathematical Theory of Communication”, *Bell System Technical Journal*, VOL 27, pp 379-423, 623-656, October 1948
- [6] 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
- [7] Howell, T.D., “Statistical properties of selected recording codes”, *IBM Journal of Research and Development*, Volume: 33, Issue: 1, 1989
- [8] P. Franzaszek, “A general method for channel coding”, *IBM J. Res. Develop.*, vol. 24, pp. 638-691, 1980
- [9] Ashley, J.J., Siegel, P.H., “A note on the Shannon capacity of run-length-limited codes”, *Information Theory, IEEE Transactions*, Volume:33, Issue:4, July 1987

- [10] Brian Marcus, Ronny Roth, Paul Siegel, "Introduction to Coding for Constrained Systems", *Lecture Notes/University of British Columbia*, 2000
- [11] Kees A. Schouhamer Immink, "Codes for Mass Data Storage Systems", 2004
- [12] S. Lin, D. J. Costello, "Error control coding: fundamentals and applications", *Prentice-Hall*, 2e, June 2004
- [13] T. Cover, J. Thomas, "Elements of Information Theory", 2e, *Wiley Series in Telecommunications and Signal Processing*, July 2006
- [14] Δ. Μπερτσεκός, Ι. Τσιτσικλής, "Εισαγωγή στις πιθανότητες", *Εκδόσεις Τζιόλας*, 2010
- [15] R.S. Varga, "Matrix Iterative Analysis", *Prentice-Hall*, 1962
- [16] Erozan M. Kurtas, Bane Vasic, "Advanced Error Control Techniques for Data Storage Systems", 2005
- [17] 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