Πολυτεχνείο Κρήτης
Τμήμα Ηλεκτρονικών Μηχανικών & Μηχανικών Υπολογιστών

Τεχνικές και Αλγόριθμοι για τη Διόρθωση Λαθών στη Μετάδοση Πληροφορίας

Καψαλάκη Σταυρούλα

Εξεταστική επιτροπή:

Καθηγητής Κ. Καλαϊτζάκης (Επιβλέπων)

Καθηγητής Γ. Σταυρακάκης

Αναπληρωτής Καθηγητής Γ. Σταμούλης

Χανιά, Σεπτέμβριος 2003

Η εργασία αφιερώνεται

στη γιαγιά μου τη Σοφία και

στην οικογένειά μου

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ.Κ. Καλαϊτζάκη για την καθοδήγηση του και την απεριόριστη υπομονή που έδειξε καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής εργασίας. Επίσης θα ήθελα να ευχαριστήσω προκαταβολικά τους καθ. Γ. Σταυρακάκη και αναπλ. καθ. Γ. Σταμούλη για το χρόνο που αφιέρωσαν στην ανάγνωση της εργασίας.

Επίσης θα ήθελα να ευχαριστήσω τον κ. Ευτύχη Κουτρούλη για την βοήθεια του όσον αφορά τους εργαστηριακούς ελέγχους του πρωτοκόλλου. Ακόμη ευχαριστώ τον Στέφανο Καρασαββίδη για την πολύτιμη βοήθεια που μου παρείχε. Ευχαριστώ τον Ιωάννη Καραμπάτσο για την υποστήριξή του όλο αυτό το διάστημα. Τέλος θέλω να ευχαριστήσω την οικογένεια μου για την υπομονή, την αγάπη και την πολύτιμη υποστήριξη τους όλα αυτά τα χρόνια.

Περιεχόμενα

Εισαγωγή

Σε όλα τα κανάλια μετάδοσης πληροφορίας παρουσιάζεται το φαινόμενο της τυχαίας παραμόρφωσης της πληροφορίας λόγω θορύβου. Ο πρωταρχικός στόχος που τίθεται στο σχεδιασμό των συστημάτων επικοινωνίας είναι η επίτευξη της μετάδοσης της πληροφορίας χωρίς λάθη. Για το λόγο αυτό έχουν αναπτυχθεί μέθοδοι κωδικοποίησης για τον έλεγχο λαθών που χρησιμοποιούνται σε όλες τις σύγχρονες υλοποιήσεις των αποθηκευτικών συστημάτων και των συστημάτων επικοινωνίας και έχουν στόχο τους να προστατεύουν τα δεδομένα από λάθη και να τα μεταδίδουν με αξιοπιστία.

Η κωδικοποίηση για τον έλεγχο και τη διόρθωση λαθών είναι η βάση του φυσικού επιπέδου των συστημάτων επικοινωνίας. Τα άλλα τμήματα σχεδιασμού ενός συστήματος επικοινωνίας όπως η εκτίμηση των χαρακτηριστικών του καναλιού ή η αποδιαμόρφωση, στόχο τους έχουν να παρέχουν στον αποκωδικοποιητή την απαραίτητη μεταδιδόμενη πληροφορία. Η κατανόηση της κωδικοποίησης ελέγχου λαθών διευκολύνει στην κατανόηση ολόκληρου του συστήματος. Οι τεχνικές και οι αλγόριθμοι στη κωδικοποίησης ελέγχου λαθών μπορούν να εφαρμοστούν και σε άλλα τμήματα ενός συστήματος επικοινωνίας (για παράδειγμα ένας βέλτιστος εξισοροπητής (equalizer) είναι παρόμοιος με ένα βέλτιστο συνελικτικό αποκωδικοποιητή) καθώς και σε κάθε είδους επικοινωνίας ασύρματη ή ενσύρματη. Επίσης με το να εκτελούνται έλεγχοι λαθών περιορίζεται σε μεγάλο βαθμό ο αριθμός των επαναμεταδόσεων και μεγιστοποιείται έτσι η συνολική απόδοση του συστήματος και ικανοποιούνται οι απαιτήσεις για την ποιότητα της μετάδοσης.

Το πρόβλημα που προκύπτει για τον έλεγχο και τη διόρθωση λαθών, δεν είναι η έλλειψη αποτελεσματικών μεθόδων κωδικοποίησης, αλλά η έλλειψη αποτελεσματικών μεθόδων με μικρή πολυπλοκότητα.

Για την επίλυση αυτού του προβλήματος πρέπει να εξοπλίσουμε τον κώδικα με περισσότερες μαθηματικές δομές. Αυτός είναι και ο μοναδικός τρόπος για να απλοποιηθούν οι διαδικασίες κωδικοποίησης και αποκωδικοποίησης.

Υπάρχουν δύο τύποι κωδίκων που χρησιμοποιούνται ευρέως για τον έλεγχο και τη διόρθωση λαθών. Στην πρώτη κατηγορία ανήκουν οι κώδικες ομάδας (block codes) όπως είναι ο Hamming κώδικας, BCH κώδικες και RS κώδικες. Στο δεύτερο τύπο εντάσσονται οι συνελικτικοί κώδικες

(convolutional codes). Οι κώδικες αυτοί επιβεβαιώνουν τον Shannon στο γεγονός ότι είναι δυνατή η μετάδοση πληροφορίας χωρίς την παρουσία λαθών.

Στη παρούσα διπλωματική εργασία τα θέματα που εξετάζονται είναι

- Η μαθηματική μοντελοποίηση ενός συστήματος επικοινωνίας.
- Η μελέτη όλων των μεθόδων, γραμμικών και κυκλικών κωδίκων, για όσο περισσότερο απλούστερη κωδικοποίηση - αποκωδικοποίηση.
- Το όριο βελτίωσης ενός κώδικα όσον αφορά το μέγεθος της προς μετάδοση πληροφορίας.
- Η απόδοση της κάθε μεθόδου, η αξιολόγηση ως προς την απόδοση τους και η ευκολία υλοποίησης της κωδικοποίησης και αποκωδικοποίησης.

Διάρθρωση εργασίας

Το κείμενο της διπλωματικής εργασίας έχει την εξής διάρθρωση:

Στο **Κεφάλαιο 1** γίνεται μια εισαγωγή στους τύπους των κωδίκων που χρησιμοποιούνται στις επικοινωνίες, στους τύπους λαθών και γενικότερα στις στρατηγικές που ακολουθούνται για τον έλεγχο λαθών.

Στο **Κεφάλαιο 2** παρουσιάζονται τα πεδία Galois που παίζουν πολύ σημαντικό ρόλο στη θεωρία κωδικοποίησης και χρησιμοποιούνται ευρέως στα ψηφιακά συστήματα μετάδοσης της πληροφορίας.

Στο **Κεφάλαιο 3** περιγράφονται αναλυτικά οι γραμμικοί κώδικες ομάδας με σύμβολα από το δυαδικό πεδίο GF(2).

Στο **Κεφάλαιο 4** Περιγράφονται οι κυκλικοί κώδικες μια πολύ σημαντική υποκατηγορία των γραμμικών κωδίκων.

Στο **Κεφάλαιο 5** περιγράφεται η αποκωδικοποίηση παγίδευσης λαθών για τους κυκλικούς κώδικες.

Στο **Κεφάλαιο 6** παρουσιάζονται αναλυτικά οι BCH (κυκλικοί) κώδικες οι οποίοι και χρησιμοποιούνται στην κωδικοποίηση – αποκωδικοποίηση του πρωτοκόλλου επικοινωνίας που υλοποιήθηκε στην παρούσα διπλωματική εργασία.

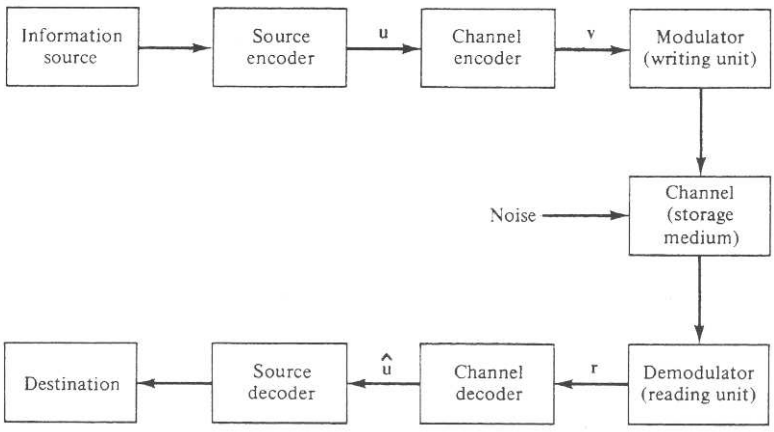
Στο **Κεφάλαιο 7** παρουσιάζεται μία υλοποίηση ενός BCH κώδικα σε συνδυασμό με μία απλή εφαρμογή επικοινωνίας με RS232 πρωτόκολλο.

ΚΕΦΑΛΑΙΟ 1

1.1 Εισαγωγή

Πρόσφατες εξελίξεις έχουν συνεισφέρει στην επίτευξη της αξιοπιστίας που απαιτείται στα ψηφιακά συστήματα και η χρήση κωδικοποίησης για τον έλεγχο των λαθών αποτελεί ένα ολοκληρωμένο τμήμα του σχεδιασμού των σύγχρονων συστημάτων επικοινωνίας.

Ένα αντιπροσωπευτικό σύστημα μετάδοσης δεδομένων παρουσιάζεται στο διάγραμμα του σχήματος 1.1. Η έξοδος της πηγής πληροφορίας η οποία πρόκειται να μεταδοθεί στον προορισμό μπορεί να είναι είτε μια συνεχής κυματομορφή είτε μια σειρά από διακριτά σύμβολα.



Σχήμα1.1 Μπλοκ διάγραμμα ενός αντιπροσωπευτικού κυκλώματος μετάδοσης δεδομένων

Ο κωδικοποιητής της πηγής μετασχηματίζει την έξοδο της πηγής σε μια σειρά από δυαδικά ψηφία που καλείται ακολουθία πληροφορίας **u**. Ο κωδικοποιητής καναλιού μετασχηματίζει τη ακολουθία πληροφορίας **u** σε μια διακριτή κωδικοποιημένη σειρά **v** που ονομάζεται κωδική λέξη. Στη συνέχεια ο διαμορφωτής μετασχηματίζει κάθε σύμβολο που είναι η έξοδος του κωδικοποιητή καναλιού σε μια κυματομορφή διάρκειας T δευτερολέπτων που είναι κατάλληλη για μετάδοση. Η κυματομορφή εισάγεται στο κανάλι και αλλοιώνεται από το θόρυβο. Στη συνέχεια ο αποδιαμορφωτής επεξεργάζεται κάθε λαμβανόμενη κυματομορφή διάρκειας T και παράγει μια

έξοδο που αντιστοιχεί στην κωδικοποιημένη σειρά **v** και ονομάζεται λαμβανόμενη ακολουθία **r**. Ο αποκωδικοποιητής του καναλιού μετασχηματίζει τη λαμβανόμενη ακολουθία **r** σε μια δυαδική σειρά που ονομάζεται εκτιμώμενη ακολουθία. Η στρατηγική αποκωδικοποίησης βασίζεται στους κανόνες κωδικοποίησης του καναλιού και στα χαρακτηριστικά του θορύβου του καναλιού. Στην ιδανική περίπτωση η εκτιμώμενη ακολουθία θα είναι ένα ακριβές αντίγραφο της ακολουθίας πληροφορίας **u**, παρόλο που ο θόρυβος μπορεί να προκαλέσει κάποια λάθη αποκωδικοποίησης. Ο αποκωδικοποιητής της πηγής μετασχηματίζει την εκτιμώμενη ακολουθία σε μια εκτίμηση της εξόδου της πηγής και την παραδίδει στον προορισμό. Το βασικό θέμα σε αυτό το σημείο είναι ο σχεδιασμός και η υλοποίηση του καναλιού με τον κωδικοποιητή/ αποκωδικοποιητή τέτοιο ώστε 1) η πληροφορία να μεταδίδεται όσο γρηγορότερα γίνεται 2) να γίνεται αξιόπιστη αναπαραγωγή της πληροφορίας που λαμβάνεται στην έξοδο του αποκωδικοποιητή καναλιού και 3) το κόστος της υλοποίησης του κωδικοποιητή και του αποκωδικοποιητή να βρίσκεται μέσα σε αποδεκτά όρια.

1.2 Τύποι κωδίκων

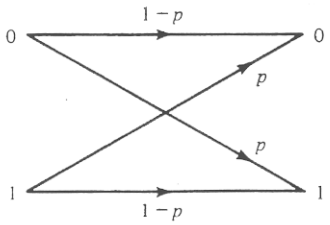
Υπάρχουν δύο τύποι κωδίκων που χρησιμοποιούνται ευρέως, οι κώδικες ομάδας και οι συνελικτικοί κώδικες. Ο κωδικοποιητής ενός κώδικα ομάδας διαιρεί την ακολουθία της πληροφορίας σε ομάδες μηνυμάτων κ ψηφίων πληροφορίας το καθένα. Μία ομάδα μηνύματος αναπαριστάται με ένα δυαδικό κ –tuple $\mathbf{u}=(u_1,u_2,\ldots,u_\kappa)$ που καλείται μήνυμα. Υπάρχουν συνολικά 2^κ διαφορετικά πιθανά μηνύματα. Ο κωδικοποιητής μετασχηματίζει ανεξάρτητα κάθε μήνυμα **u** σε ένα n – tuple $v = (v_1, v_2,\ldots, v_n)$ διακριτών συμβόλων που καλείται κωδική λέξη. Επομένως υπάρχουν 2^κ διαφορετικές κωδικές λέξεις που αντιστοιχούν στα 2^κ διαφορετικά μηνύματα στην έξοδο του κωδικοποιητή. Αυτό το σύνολο των κωδικών λέξεων μήκους n καλείται (n,k) κώδικας ομάδας. Το κλάσμα R = k/n ονομάζεται ρυθμός κώδικα και μπορεί να ερμηνευθεί ως ο αριθμός ψηφίων πληροφορίας που εισέρχονται στον κωδικοποιητή ανά μεταδιδόμενο σύμβολο. Από τη στιγμή που η εξαγόμενη κωδική λέξη (με τα n σύμβολα) εξαρτάται μόνο από το αντίστοιχο μήνυμα εισόδου των κ bits ο κωδικοποιητής δεν χρειάζεται να διαθέτει μνήμη και μπορεί να υλοποιηθεί με ένα συνδυαστικό λογικό κύκλωμα.

Σε ένα δυαδικό κώδικα, κάθε κωδική λέξη **v** είναι επίσης δυαδική. Επομένως για να είναι ένας δυαδικός κώδικας χρήσιμος $\kappa \leq n$ ή $R \leq 1$. Όταν $\kappa < n$, $n - k$ ψηφία πλεονασμού μπορούν να προστεθούν σε ένα μήνυμα για να διαμορφώσουν μια κωδική λέξη. Αυτά τα ψηφία πλεονασμού δίνουν τη δυνατότητα να αντιμετωπίζεται ο θόρυβος του καναλιού. Η επιλογή αυτών των ψηφίων

πλεονασμού για την επίτευξη αξιόπιστης μετάδοσης μέσα από ένα κανάλι με θόρυβο είναι το βασικό πρόβλημα σχεδιασμού ενός κωδικοποιητή. Ο κωδικοποιητής ενός συνελικτικού κώδικα δέχεται και αυτός μία ομάδα των κ ψηφίων από την ακολουθία πληροφορίας **u** και παράγει μια κωδικοποιημένη ακολουθία (κωδική λέξη) **v,** μία ομάδα των n συμβόλων. Όμως κάθε κωδικοποιημένη ομάδα εξαρτάται όχι μόνο από την αντίστοιχη ομάδα μηνύματος των κ bits της ίδιας χρονικής στιγμής, αλλά επίσης και από τις m προηγούμενες ομάδες μηνύματος. Δηλαδή ο κωδικοποιητής έχει μνήμη τάξης m. Οι συνελικτικοί κώδικες δεν θα μας απασχολήσουν στην παρούσα διπλωματική εργασία.

1.3 Τύποι λαθών

Σε κανάλια χωρίς μνήμη, ο θόρυβος επηρεάζει κάθε μεταδιδόμενο σύμβολο ανεξάρτητα. Για παράδειγμα, θεωρούμε ένα δυαδικό συμμετρικό κανάλι που το διάγραμμα μετάδοσης του φαίνεται στο σχήμα 1.2.



Σχήμα 1.2 Διάγραμμα πιθανότητας μετάδοσης σε ένα συμμετρικό δυαδικό κανάλι

Κάθε μεταδιδόμενο ψηφίο έχει πιθανότητα ρ να μεταδοθεί λανθασμένα και πιθανότητα 1 – ρ να μεταδοθεί σωστά ανεξάρτητα από τα άλλα μεταδιδόμενα ψηφία. Άρα τα λάθη μετάδοσης συμβαίνουν τυχαία στην λαμβανόμενη ακολουθία, και έτσι τα κανάλια χωρίς μνήμη καλούνται κανάλια τυχαίων λαθών. Οι κώδικες που αναλαμβάνουν τη διόρθωση τυχαίων λαθών ονομάζονται κώδικες διόρθωσης τυχαίων λαθών. Όλοι οι κώδικες που θα εξετάσουμε στα επόμενα κεφάλαια ανήκουν σε αυτή την κατηγορία. Υπάρχει και η κατηγορία των λαθών «ξεσπάσματος » που επειδή αναφέρονται σε περιπτώσεις καναλιών με μνήμη δεν θα μας απασχολήσουν στο σημείο αυτό.

1.4 Στρατηγικές ελέγχου λαθών

Υπάρχουν δύο κατηγορίες συστημάτων μετάδοσης πληροφορίας τα μονόδρομα και τα αμφίδρομα συστήματα μετάδοσης πληροφορίας. Στην περίπτωση των μονόδρομων συστημάτων μετάδοσης πληροφορίας, η πληροφορία μεταδίδεται αυστηρά προς μια κατεύθυνση από τον αποστολέα στον παραλήπτη. Στην περίπτωση των αμφίδρομων συστημάτων πληροφορίας που είναι και η πιο συνηθισμένη η πληροφορία μπορεί να μεταδοθεί και προς τις δύο κατευθύνσεις και ο αποστολέας να λειτουργεί και σαν παραλήπτης και αντίστροφα. Ο έλεγχος λαθών για ένα αμφίδρομο σύστημα μπορεί να επιτευχθεί με την ανίχνευση λαθών και την επαναμετάδοση της πληροφορίας, που ονομάζεται αυτόματη αίτηση επανάληψης (ARQ). Σε αυτά τα συστήματα όταν ανιχνεύονται λάθη από τον παραλήπτη, μια αίτηση στέλνεται στον αποστολέα για να επαναλάβει το μήνυμα, και αυτό συνεχίζεται έως ότου το λάβει σωστά.

Υπάρχουν δύο κατηγορίες συστημάτων με αυτόματη αίτηση επανάληψης : Παύση και αναμονή για αυτόματη αίτηση επανάληψης, και συνεχής αυτόματη αίτηση επανάληψης. Στην πρώτη περίπτωση ο αποστολέας στέλνει μια κωδική λέξη στον παραλήπτη και περιμένει για μια επιβεβαίωση λήψης (ACK) ή μια αρνητική επιβεβαίωση (NACK) από τον παραλήπτη. Αν παραλάβει την ACK(δεν έχουν ανιχνευθεί λάθη) τότε στέλνει την επόμενη κωδική λέξη. Αν παραλάβει τη NACK (έχουν ανιχνευθεί λάθη) τότε ξαναστέλνει την ίδια κωδική λέξη και αυτό γίνεται μέχρι να λάβει τη σωστή κωδική λέξη.

Στην δεύτερη περίπτωση ο αποστολέας στέλνει συνεχώς κωδικές λέξεις στον παραλήπτη και παραλαμβάνει συνεχώς ACK. Αν παραλάβει κάποια NACK ο αποστολέας ξεκινάει την επαναμετάδοση της κωδικής λέξης που παραλήφθηκε λανθασμένα και όλων των κωδικών λέξεων που στάλθηκαν μετά από αυτή. Σε αυτή τη περίπτωση χρησιμοποιούνται πρωτόκολλα όπως τα Go –back-N και selective- repeat.

Εκτός όμως από τις παραπάνω περιπτώσεις υπάρχει και η δυνατότητα η περίπτωση λάθους να μην αντιμετωπίζεται με ανίχνευση και επαναμετάδοση της λανθασμένης κωδικής λέξης αλλά ο παραλήπτης να είναι εφοδιασμένος με ειδικούς κώδικες διόρθωσης λαθών οι οποίοι αυτόματα θα διορθώνουν τα λάθη και δεν θα χρειάζεται επαναμετάδοση της κωδικής λέξης.

Το βασικό πλεονέκτημα των δύο πρώτων τεχνικών είναι ότι με την ανίχνευση των λαθών απαιτείται απλούστερη κωδικοποίηση από ότι στη διόρθωση των λαθών. Όταν όμως η μετάδοση

πληροφορίας εκτελείται σε ένα περιβάλλον με υψηλό θόρυβο οι επαναμεταδόσεις μπορούν να γίνονται τόσο συχνά ώστε η συνολική απόδοση του συστήματος να μειώνεται αισθητά και η ταχύτητα με την οποία να λαμβάνονται ορθά οι κωδικές λέξεις, να είναι πολύ χαμηλότερη από τις δύο πρώτες περιπτώσεις απλής ανίχνευσης λαθών τότε πρέπει σαφώς να επιλέγεται η διόρθωση λαθών.

ΚΕΦΑΛΑΙΟ 2 - GALOIS FIELD GF(2^m) - ΔΙΑΝΥΣΜΑΤΙΚΟΙ ΧΩΡΟΙ

2.1 Δομή του Galois Field GF(2^m)

Σ΄ αυτό το σημείο γίνεται παρουσίαση της μεθόδου που ακολουθείται για την κατασκευή του Galois Field GF(2^m) των 2^m στοιχείων (m>1) από το δυαδικό πεδίο GF(2). Ξεκινάμε με τα δύο στοιχεία 0,1 του GF(2) και ένα καινούργιο στοιχείο a.

Για τον πολλαπλασιασμό « · » ισχύουν τα εξής :

$0 \cdot 0 = 0$, $0 \cdot 1 = 1 \cdot 0 = 0$, $1 \cdot 1 = 1$,

$0 \cdot a = a \cdot 0 = 0$, $1 \cdot a = a \cdot 1 = a$

$a^2 = a \cdot a$, $a^3 = a \cdot a \cdot a$, $a^j = a \cdot a \cdot a \ldots \cdot a$ (j φορές)

Από τον ορισμό του πολλαπλασιασμού για τις δυνάμεις a^j θα έχουμε :

$0 \cdot \mathbf{a^j = a^j} \cdot 0 = 0$

$1 \cdot a^j = a^j \cdot 1 = a^j$

$a^j \cdot a^i = a^i \cdot a^j = a^{i+j}$

Με αυτό τον τρόπο ορίζουμε το σύνολο των στοιχείων που προκύπτουν από την πράξη του πολλαπλασιασμού ως εξής :

$F = \{ 0, 1, a , a^2 , a^3 ,……, a^j , ….\}$

Θέτουμε περιορισμό για το στοιχείο a έτσι ώστε το σύνολο F περιέχει μόνο 2^m στοιχεία και είναι κλειστό ως προς την πράξη του πολλαπλασιασμού . Έστω p(X) είναι ένα βασικό (primitive) πολυώνυμο βαθμού m ορισμένο στο GF(2). Υποθέτουμε ότι p(a) = 0 . Τότε το p(x) διαιρεί το

$X^{2^{m}-1} + 1$ και έτσι

Αντικαθιστώντας το X με το a :

$$a^{2^{m}-1} + 1 = q(a) \cdot p(a) .$$

και επειδή p(a) = 0

$$a^{2^{m}-1} + 1 = 0 \rightarrow a^{2^{m}-1} = 1 \text{ (modulo -2 addition)}$$

Επομένως υπό την συνθήκη ότι p(a) = 0 , το σύνολο F γίνεται πεπερασμένο και περιέχει τα εξής διακριτά στοιχεία :

$F^* = \{ 0, 1, a , a^2 , a^3 ,……, a^{2^{m}-2} \}$

Κάτι που θα πρέπει να αναφερθεί είναι ότι το αντίστροφο του a^i που ανήκει στο F είναι $a^{2^{m}-i-1}$ καθώς :

$a^i \cdot a^{2^{m-i-1}} = a^{2^{m-1}} = 1$

Το επόμενο βήμα είναι να ορίσουμε την πράξη της πρόσθεσης « + » στο F * έτσι ώστε το F * να είναι ένα κλειστό σύνολο ως προς την πράξη της πρόσθεσης.

Για $0 <= i <= 2^m - 1$, διαιρούμε το πολυώνυμο X^i με το p(X) και έχουμε :

$$X^i = q_i(X) \cdot p(X) + a_i(X) \quad (1)$$

όπου $q_i(X)$ και $a_i(X)$ είναι το πηλίκο και το υπόλοιπο αντίστοιχα . Το $a_i(X)$ είναι ένα πολυώνυμο βαθμού m – 1 ή λιγότερο ορισμένο στο GF(2) και έχει την εξής μορφή :

$$a_i(X) = a_{i0} + a_{i1} \cdot X + a_{i2} \cdot X^2 + \ldots + a_{i,m-1} \cdot X^{m-1} \quad (1)$$

Από τη στιγμή που τα X , p(X) δεν έχουν κανένα κοινό παράγοντα εκτός του 1 , το X^i δεν διαιρείται ακριβώς με το p(X) . Επομένως για κάθε i>=0 το $a_i(X)$ δεν είναι μηδέν (2).

Επίσης αποδεικνύεται ότι για κάθε i, $j < 2^m - 1$ και για $i \neq j$ τότε και το $ai(x) \neq aj(x)$. Επομένως για κάθε i= 0,1,2,…,$2^m - 2$ παίρνουμε $2^m - 1$ διακριτά μη μηδενικά πολυώνυμα $a_i(x)$ βαθμού m –1 ή μικρότερου. Έτσι αντικαθιστώντας το X με α στην σχέση (1) και χρησιμοποιώντας την ισότητα $q_i(a) * 0 = 0$ παίρνουμε την παρακάτω πολυωνυμική έκφραση για το a^i :

$$a^i = a_i(a) = a_{i0} + a_{i1} \cdot a + a_{i2} \cdot a^2 + \ldots + a_{i,m-1} \cdot a^{m-1}$$

Επομένως βλέπουμε ότι τα $2^m - 1$ με μηδενικά στοιχεία $\alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{2^{m-2}}$ στο F* παριστάνονται με $2^m - 1$ διακριτά μη μηδενικά πολυώνυμα του α του GF(2) βαθμού m-1 ή μικρότερου. Το μηδενικό στοιχείο 0 του F* μπορεί να αναπαρασταθεί με το μηδενικό πολυώνυμο.

Η πράξη της πρόσθεσης « + » στο F* ορίζεται ως εξής :

0+0=0

και για $0 \le i, j < 2^m - 1$,

$$0 + a^i = a^i + 0 = ai,$$

$$a^i + a^j = (a_{i0} + a_{i1} \cdot a + a_{i2} \cdot a^2 + \ldots + a_{i,m-1} \cdot a^{m-1}) + (a_{jo} + a_{j1} \cdot a + a_{j2} \cdot a^2 + \ldots + a_{j,m-1} \cdot a^{m-1})$$

$$= (a_{i0} + a_{jo}) + (a_{i1} + a_{j1}) \cdot a + \ldots + (a_{i,m-1} + a_{j,m-1}) \cdot a^{m-1}$$

όπου το $a_{i,l} + a_{j,l}$ υπολογίζεται με modulo –2 πρόσθεση .

Εάν τα i=j τότε :

$$a^i + a^j = 0 \quad (2)$$

και για $i \neq j$

$$(a_{i0} + a_{jo}) + (a_{i1} + a_{j1}) \cdot a + \ldots + (a_{i,m-1} + a_{j,m-1}) \cdot a^{m-1}$$

είναι μη μηδενικό και πρέπει να είναι η πολυωνυμική έκφραση κάποιου a^k που ανήκει στο F*. Το σύνολο F* είναι ένα σύνολο κλειστό ως προς την πράξη της πρόσθεσης « + » όπως αυτή ορίστηκε παραπάνω. Μπορούμε εύκολα να επαληθεύσουμε ότι το μηδενικό στοιχείο του συνόλου F* είναι το μηδέν. Επίσης από τη σχέση (2) μπορούμε να ορίσουμε ως αντίθετο κάθε στοιχείου του συνόλου F* τον εαυτό του. Χρησιμοποιώντας το γεγονός ότι η modulo-2 πρόσθεση είναι μια πράξη αντιμεταθετική και προσεταιριστική επόμενο είναι ότι και η πράξη «+» στο F* θα είναι μια πράξη αντιμεταθετική και προσεταιριστική.

Μέχρι τώρα έχει αποδειχτεί ότι το σύνολο F* = {0,1,a,a^2,a^3,..., $a^{2^{m-2}}$} είναι ένα σύνολο στο οποίο ισχύει η αντιμεταθετική ιδιότητα όσον αφορά την πράξη της πρόσθεσης και τα μη μηδενικά στοιχεία του F* είναι ένα σύνολο στο οποίο ισχύει η αντιμεταθετική ιδιότητα όσον αφορά την πράξη του πολλαπλασιασμού. Μπορούμε εύκολα να διαπιστώσουμε ότι ο πολλαπλασιασμός στο F* είναι πράξη επιμεριστική ως προς την πρόσθεση στο F*. Επομένως το σύνολο

F* = {0,1,a,a^2,a^3,..., $a^{2^m -2}$} είναι ένα πεδίο Galois 2^m στοιχείων ,GF(2^m) . Επίσης παρατηρούμε ότι η πρόσθεση και ο πολλαπλασιασμός που ορίζονται στο F*= GF(2^m) είναι η modulo-2 πρόσθεση και πολλαπλασιασμός. Επομένως το υποσύνολο {0,1} είναι ένα υποπεδίο του GF(2^m) (δηλαδή το GF(2) είναι ένα υποπεδίο του GF(2^m)). Το δυαδικό πεδίο GF(2) συνήθως ονομάζεται πεδίο βάση (ground field)του GF(2^m). Η χαρακτηριστική του GF(2^m) είναι 2.

Στην διαδικασία δόμησης του GF(2^m) από το GF(2) έχουν αναπτυχθεί δύο αναπαραστάσεις για τα μη μηδενικά στοιχεία του GF(2^m):η αναπαράσταση με δυνάμεις και η αναπαράσταση με πολυώνυμα. Η αναπαράσταση με δυνάμεις χρησιμοποιείται στον πολλαπλασιασμό και η αναπαράσταση με πολυώνυμα χρησιμοποιείται στην πρόσθεση.

Υπάρχει άλλος ένας χρήσιμος τρόπος αναπαράστασης των στοιχείων του πεδίου GF(2^m). Έστω $\alpha_0+\alpha_1 \cdot a + \alpha_2 \cdot a^2+ \ldots+ \alpha_{m-1} \cdot a^{m-1}$ η πολυωνυμική αναπαράσταση του στοιχείου β του πεδίου. Μπορούμε να αναπαραστήσουμε το β με μία ταξινομημένη σειρά m στοιχείων η οποία ονομάζεται m-tuple και είναι της μορφής $(\alpha_0,\alpha_1,\alpha_2,...,\alpha_{m-1})$ όπου τα m στοιχεία είναι οι συντελεστές της πολυωνυμικής αναπαράστασης του β. Βλέπουμε λοιπόν ότι υπάρχει μια προς μία αντιστοιχία μεταξύ του m-tuple και της πολυωνυμικής αναπαράστασης του β . Το μηδενικό στοιχείο του GF(2^m) αναπαρίσταται από το μηδενικό m-tuple (0,0,0,...,0).

Έστω $(b_0,b_1,b_2,\ldots,b_{m-1})$ είναι η m-tuple αναπαράσταση του στοιχείου γ του GF(2^m). Για να προσθέσουμε τα β, γ απλά προσθέτουμε τα αντίστοιχα στοιχεία των m-tuple αντίστοιχα :

$$(\alpha_0 +b_0, \alpha_1+b_1, \alpha_2+ b_2 ,...,\alpha_{m-1}+ b_{m-1}) ,$$

όπου $\alpha_i +b_i$ υπολογίζεται με modulo-2 πρόσθεση.

ΠΙΝΑΚΑΣ 1 Τρεις αναπαραστάσεις για τα στοιχεία του GF(2^4) που παράγονται από $P(X)=1+X+X^4$

Αναπαράσταση με δυνάμεις	Πολυωνυμική αναπαράσταση	4-tuple αναπαράσταση
0	0	(0000)
1	1	(1000)
α	α	(0100)
α^2	α^2	(0010)
α^3	α^3	(0001)
α^4	$1+\alpha$	(1100)
α^5	$\alpha+\alpha^2$	(0110)
α^6	$\alpha^2+\alpha^3$	(0011)
α^7	$1+\alpha+\alpha^3$	(1101)
α^8	$1+\alpha^2$	(1010)
α^9	$\alpha+\alpha^3$	(0101)
α^{10}	$1+\alpha+\alpha^2$	(1110)
α^{11}	$\alpha+\alpha^2+\alpha^3$	(0111)
α^{12}	$1+ \alpha+\alpha^2+\alpha^3$	(1111)
α^{13}	$1+ \alpha^2+\alpha^3$	(1011)
α^{14}	$1+\alpha^3$	(1001)

2.2 Βασικές ιδιότητες του πεδίου Galois

Στο σημείο αυτό παραθέτονται κάποιες πολύ βασικές ιδιότητες και θεωρήματα που ισχύουν για το πεδίο Galois.

1^{o} θεώρημα : έστω πολυώνυμο f(x) με συντελεστές από το GF(2). Έστω β είναι στοιχείο ενός πεδίου που είναι προέκταση του GF(2). Εάν το β είναι ρίζα του f(x) τότε για κάθε $l \geq 0$ $\beta^{2^{\wedge}l}$ είναι επίσης ρίζα του f(x).

Το στοιχείο $\beta^{2^{\wedge}1}$ ονομάζεται συζυγές του β. Το θεώρημα αυτό είναι πολύ χρήσιμο όσον αφορά την εύρεση ριζών του πολυωνύμου f(x). Σύμφωνα με το θεώρημα αν β είναι ρίζα του f(x) τότε όλες οι διακριτές συζυγείς δυνάμεις του β [που είναι στοιχεία του $GF(2^{m})$] είναι επίσης ρίζες του f(x).

Έστω ότι το β είναι ένα μη μηδενικό στοιχείο του $GF(2^{m})$. Από το προηγούμενο θεώρημα προκύπτει ότι : $\beta^{2^{\wedge}m-1} = 1$.

Προσθέτοντας το 1 και στα δύο μέλη της παραπάνω σχέσης προκύπτει ότι :

$$\beta^{2^{\wedge}m-1} + 1 = 0$$

Από τη σχέση αυτή φαίνεται ότι το β είναι ρίζα του πολυωνύμου $X^{2^{\wedge}m-1} + 1 = 0$. Επομένως κάθε στοιχείο του $GF(2^{m})$ είναι ρίζα του $X^{2^{\wedge}m-1} + 1$.

Από τα παραπάνω προκύπτει το **2^{o} θεώρημα** σύμφωνα με το οποίο : Τα $2^{m} - 1$ μη μηδενικά στοιχεία του $GF(2^{m})$ διαμορφώνουν όλες τις ρίζες του $X^{2^{\wedge}m-1} + 1$.

Από τη στιγμή που το μηδενικό στοιχείο του $GF(2^{m})$ είναι η ρίζα του X από το 2^{o} θεώρημα προκύπτει το πόρισμα ότι :

Τα στοιχεία του $GF(2^{m})$ διαμορφώνουν όλες τις ρίζες του $X^{2^{\wedge}m} + X$.

Από τη στιγμή που κάθε στοιχείο β του $GF(2^{m})$ είναι ρίζα του πολυωνύμου $X^{2^{\wedge}m} + X$,το β μπορεί να είναι ρίζα ενός πολυωνύμου ορισμένο στο GF(2) με βαθμό μικρότερο του 2^{m}. Έστω φ(X) είναι το πολυώνυμο μικρότερου βαθμού ορισμένο στο GF(2) τέτοιο ώστε φ(β) = 0[αποδεικνύεται εύκολα ότι το πολυώνυμο αυτό είναι μοναδικό]. Το πολυώνυμο αυτό φ(X) ονομάζεται ελάχιστο πολυώνυμο του β (minimal polynomial of β).

Για το ελάχιστο πολυώνυμο ισχύουν τα εξής θεωρήματα :

Θεώρημα (3) . Το minimal polynomial φ(X) του στοιχείου β είναι irreducible .

Θεώρημα (4). Έστω f(X) πολυώνυμο ορισμένο στο GF(2). Έστω φ(X) το minimal polynomial του στοιχείου β του πεδίου. Εάν β είναι ρίζα του f(X), τότε το f(X) διαιρείται με το φ(X).

Θεώρημα (5) Το minimal polynomial φ(X) ενός στοιχείου β του $GF(2^m)$ διαιρεί το $X^{2^m}+X$.

Θεώρημα (6) Έστω f(X) είναι ένα irreducible πολυώνυμο του GF(2). Έστω β στοιχείο του $GF(2^m)$. Έστω φ(X) το ελάχιστο πολυώνυμο του β. Εάν f(β) =0 τότε f(X) = φ(X) .

Θεώρημα (7) Αν β είναι ένα στοιχείο του $GF(2^m)$ και e ο μικρότερος μη αρνητικός ακέραιος τέτοιος ώστε $\beta^{2^e} = \beta$ τότε

$$f(X) = \prod_{i=0}^{e-1} (X + \beta^{2^i})$$

είναι ένα irreducible πολυώνυμο στο GF(2).

Μία άμεση συνέπεια των θεωρημάτων 6,7 είναι το θεώρημα 8 σύμφωνα με το οποίο αν το φ(X) είναι το ελάχιστο πολυώνυμο του στοιχείου β στο $GF(2^m)$ και το e είναι ο μικρότερος ακέραιος τέτοιος ώστε $\beta^{2^e} = \beta$ τότε :

$$\varphi(X) = \prod_{i=0}^{e-1} (X + \beta^{2^i})$$

Πίνακας 2 Τα ελάχιστα πολυώνυμα των στοιχείων του $GF(2^4)$ που παράγονται από το

$\rho(X)= X^4+X+1$

Συζυγείς ρίζες	**Ελάχιστα πολυώνυμα(minimal polynomial)**
0	X
1	X+1
$\alpha,\alpha^2,\alpha^4,\alpha^8$	X^4+X+1
$\alpha^3,\alpha^6,\alpha^9,\alpha^{12}$	$X^4+X^3+X^2+X+1$
α^5,α^{10}	X^2+X+1
$\alpha^7,\alpha^{11},\alpha^{13},\alpha^{14}$	X^4+X^3+1

Θεώρημα 9 : Έστω φ(X) το ελάχιστο πολυώνυμο του στοιχείου β του πεδίου $GF(2^m)$. Έστω e ο βαθμός του πολυωνύμου φ(X). Τότε ο e είναι ο μικρότερος ακέραιος για τον οποίο $\beta^{2\hat{}e} = \beta$. Επιπλέον , $e \le m$.

Συγκεκριμένα ο βαθμός του ελάχιστου πολυώνυμου κάθε στοιχείου στο $GF(2^m)$ διαιρεί το m. Ο Πίνακας 2 δείχνει ότι ο βαθμός του ελάχιστου πολυωνύμου κάθε στοιχείου του $GF(2^4)$ διαιρεί το 4. Για την κατασκευή του Galois πεδίου $GF(2^m)$,χρησιμοποιούμε το βασικό (primitive) πολυώνυμο p(X) βαθμού m και απαιτούμε το στοιχείο a να είναι ρίζα του p(X). Από τη στιγμή που οι δυνάμεις του a παράγουν όλα τα μη μηδενικά στοιχεία του $GF(2^m)$, το a είναι το βασικό (primitive) στοιχείο. Στην πραγματικότητα κάθε συζυγής του a είναι βασικό στοιχείο του $GF(2^m)$.

Θεώρημα 10: Αν β είναι βασικό στοιχείο του $GF(2^m)$ τότε όλες οι συζυγείς τιμές του $\beta^2,\beta^{2\hat{}2},...$ είναι επίσης βασικά στοιχεία του $GF(2^m)$.

Θεώρημα 11: Αν β είναι ένα στοιχείο βαθμού n στο $GF(2^m)$,τότε όλες οι συζυγείς τιμές του είναι του ίδιου βαθμού n.

2.3 Διανυσματικοί χώροι

Έστω V ένα σύνολο στοιχείων στα οποία ορίζεται μία δυαδική πράξη που ονομάζεται πρόσθεση +. Έστω F είναι ένα πεδίο. Ορίζεται επίσης η πράξη πολλαπλασιασμού ,συμβολίζεται με (· **)**, μεταξύ των στοιχείων του F και του V. Το σύνολο V καλείται διανυσματικός χώρος εφοδιασμένος με κανόνες διανυσματικής πρόσθεσης και πολλαπλασιασμού με στοιχεία του πεδίου F εάν ικανοποιεί τις ακόλουθες συνθήκες:

1. Η πράξη της πρόσθεσης είναι αντιμεταθετική στο V.
2. Για κάθε στοιχείο a του F και για κάθε στοιχείο v του V,το a · v είναι ένα στοιχείο του V.
3. Για οποιαδήποτε στοιχεία v και u του V,και για οποιαδήποτε στοιχεία a και b του F ισχύουν τα ακόλουθα : α) a · **(u+v) = a** · u + a · v β) (a + b) · v = a · v + b · v(επιμεριστική)
4. Για κάθε στοιχείο v του V και για οποιαδήποτε στοιχεία a,b του F ισχύει ότι :

   (a · b) · v=a · (b · v).(προσεταιριστική)
5. Το στοιχείο 1 είναι το μοναδιαίο στοιχείο του F. Τότε για κάθε στοιχείο v του V, 1· v=v.

Τα στοιχεία του V ονομάζονται διανύσματα (vectors) και τα στοιχεία του F ονομάζονται βαθμωτοί (scalars). Η πρόσθεση στο V ονομάζεται πρόσθεση διανυσμάτων και ο πολλαπλασιασμός μεταξύ ενός βαθμωτού a του F και ενός διανύσματος του V που μας δίνει ένα διάνυσμα του V αναφέρεται ως βαθμωτός πολλαπλασιασμός (ή γινόμενο) . Το μηδενικό στοιχείο του V ως προς την πρόσθεση είναι το μηδενικό διάνυσμα και συμβολίζεται με **0**.

Κάποιες βασικές ιδιότητες του διανυσματικού χώρου είναι οι εξής :

I) Έστω το 0 είναι το μηδενικό στοιχείο του F. Τότε για κάθε διάνυσμα v του V , 0 ·v = **0.**

II) Για κάθε c βαθμωτό στοιχείο του F, **0** ·c = **0**.

III) Για κάθε c βαθμωτό στοιχείο του F και για κάθε διάνυσμα v του V **(-c)** · **v=c** · **(- v)= - (c** · **v)**

Από τα παραπάνω ορίζεται ότι ο αντίθετος του c·v είναι είτε ο (-c)·v ή ο c·(-v).

Στο σημείο αυτό παρουσιάζεται ένας πολύ χρήσιμος διανυσματικός χώρος ορισμένος στο GF(2) ο οποίος παίζει ένα πολύ βασικό ρόλο στην θεωρία κωδικοποίησης. Θεωρούμε μια ταξινομημένη σειρά n στοιχείων ,

$$(\alpha_0,\alpha_1,\alpha_2,...,\alpha_{n-1})$$

στην οποία κάθε στοιχείο α_i είναι ένα στοιχείο του GF(2) (δηλαδή $\alpha_i = 1$ ή $\alpha_i = 0$). Αυτή η σειρά ορίζεται ως ένα n – tuple του GF(2). Από τη στιγμή που υπάρχουν δύο πιθανές τιμές για κάθε α_i , μπορούμε να έχουμε 2^n διακριτά n – tuples. Έστω V_n είναι το σύνολο των 2^n διακριτών n – tuples. Ορίζουμε την πρόσθεση + στο V_n ως εξής : Για κάθε $u = (u_0,u_1,\ldots,u_{n-1})$ και για $v = (v_0,v_1,\ldots v_{n-1})$

$u + v = (u_0+v_0 , u_1+v_1 ,\ldots,u_{n-1}+v_{n-1})$ (1^{α})

όπου η πρόσθεση για κάθε u_i+v_i είναι μια 2-modulo πρόσθεση. Επομένως κάθε u + v είναι επίσης ένα n – tuple του GF(2). Άρα το V_n είναι ένα κλειστό σύνολο ως προς την πρόσθεση έτσι όπως αυτή ορίστηκε παραπάνω και μπορεί εύκολα να δειχτεί ότι ισχύει η αντιμεταθετική ιδιότητα για την πράξη της πρόσθεσης .

Το μηδενικό στοιχείο του V_n είναι το στοιχείο **0**= (0,0,...0) όσο αφορά τη πράξη της πρόσθεσης . Για κάθε στοιχείο v του V_n ισχύει :

$$v + v=(v_0+v_0,v_1+v_1,\ldots,v_{n-1}+v_{n-1}) =(0,0,\ldots,0)= \mathbf{0}$$

Επομένως για κάθε στοιχείο v του V_n το αντίστροφο του ,όσον αφορά την πρόσθεση, είναι ο εαυτός του. Από τη στιγμή που η 2 –modulo πρόσθεση είναι πράξη αντιμεταθετική και προσεταιριστική τότε και η πρόσθεση όπως αυτή ορίστηκε (1^{α}) είναι επίσης αντιμεταθετική και προσεταιριστική. Επομένως το V_n είναι ένα σύνολο στο οποίο ισχύει η αντιμεταθετική ιδιότητα ως προς την πράξη της πρόσθεσης.

Το επόμενο βήμα είναι ο ορισμός της πράξης του πολλαπλασιασμού ενός n-tuple v του V_n με ένα στοιχείο a από το GF(2). Για την πράξη αυτή έχουμε :

$$a\cdot(v_0,v_1,\ldots v_{n-1})= (a\cdot v_0,a\cdot v_1,\ldots,a\cdot v_{n-1}),$$

όπου το κάθε γινόμενο $a\cdot v_i$ υπολογίζεται με βάση το 2-modulo πολλαπλασιασμό. Άρα το $a\cdot(v_0,v_1,\ldots v_{n-1})$ είναι ένα n-tuple του V_n . Αν το a =1 τότε $1\cdot(v_0,v_1,\ldots v_{n-1})= (1\cdot v_0,1\cdot v_1,\ldots,1\cdot v_{n-1}) = (v_0, v_1,\ldots, v_{n-1})$.

Από τα παραπάνω καταλήγουμε ότι η πρόσθεση μεταξύ διανυσμάτων και ο βαθμωτός πολλαπλασιασμός είναι και οι δύο πράξεις επιμεριστικές και προσεταιριστικές. Το συμπέρασμα που εξάγεται είναι ότι το σύνολο V_n όλων των n – tuples ορισμένα στο GF(2)διαμορφώνουν ένα διανυσματικό χώρο με πεδίο ορισμού τοGF(2).

Θεώρημα Έστω S είναι ένα υποσύνολο του διανυσματικού χώρου V. Τότε το S είναι ένας υποχώρος του V εάν ικανοποιούνται οι παρακάτω συνθήκες :

(i) για κάθε δύο διανύσματα **v** και **u** του S, το **v+u** είναι επίσης διάνυσμα του S.

(ii) Για κάθε στοιχείο a του F και για κάθε διάνυσμα **u** του S, το a · **u** είναι επίσης στοιχείο του S.

Έστω $v_1,v_2,\ldots,v_k$ k διανύσματα του διανυσματικού χώρου V. Έστω $a_1,a_2,\ldots,a_k$ είναι k βαθμωτοί από το F. Το άθροισμα

$$a_1v_1+a_2v_2+\ldots+a_k\, v_k$$

ονομάζεται γραμμικός συνδυασμός των $v_1,v_2,\ldots,v_k$. Το άθροισμα δύο γραμμικών συνδυασμών των $v_1,v_2,\ldots,v_k$,

$$(a_1v_1+a_2v_2+\ldots+a_k\, v_k) + (b_1v_1+b_2v_2+\ldots+b_k\, v_k) = (a_1 + b1)v_1 + (a_2+ b_2)v_2 +\ldots+ (a_k+ b_k)v_k$$

είναι επίσης ένας γραμμικός συνδυασμός των $v_1,v_2,\ldots,v_k$ αλλά και το γινόμενο ενός βαθμωτού c του F και ενός γραμμικού συνδυασμού των $v_1,v_2,\ldots,v_k$, είναι ένας επίσης γραμμικός συνδυασμός των $v_1,v_2,\ldots,v_\kappa$,

$$c\cdot(a_1v_1+a_2v_2+\ldots+a_k\, v_k) = (c\cdot a_1)v_1+(c\cdot a_2)v_2+\ldots+(c\cdot a_k)\, v_k$$

Επομένως καταλήγουμε στο **θεώρημα** ότι: Έστω $v_1,v_2,\ldots,v_k$ k διανύσματα του διανυσματικού χώρου V. Το σύνολο όλων των γραμμικών συνδυασμών των $v_1,v_2,\ldots,v_k$ αποτελεί ένα υπόχωρο του διανυσματικού χώρου V.

Για το σύνολο των διανυσμάτων $v_1,v_2,\ldots,v_k$ ισχύει ότι αν για $\alpha_1,\alpha_2,...,\alpha_\kappa$ του F το άθροισμα

$\alpha_1 v_1+\alpha_2 v_2+\ldots+\alpha_\kappa v_k = 0$ τότε τα διανύσματα είναι *γραμμικώς εξαρτημένα* διαφορετικά αν

$\alpha_1 v_1+\alpha_2 v_2+\ldots+\alpha_\kappa v_k \neq 0$ τότε τα διανύσματα είναι *γραμμικώς ανεξάρτητα.*

Ένα σύνολο διανυσμάτων ορίζουν ένα διανυσματικό χώρο V αν κάθε διάνυσμα του χώρου μπορεί να γραφεί ως γραμμικός συνδυασμός τους . Τα διανύσματα αυτά καλούνται *βάση* του χώρου και ο αριθμός των διανυσμάτων αποτελούν την *διάσταση* του χώρου.

Από τη στιγμή που τα $\alpha_1,\alpha_2,..\alpha_\kappa$ μπορούν να πάρουν δύο τιμές 0 και 1 υπάρχουν 2^κ διακριτά διανύσματα στο χώρο V που είναι ο υπόχωρος του V_n.

Έστω $u= (u_0,u_1,\ldots,u_{n-1})$ και $v=(v_0,v_1,\ldots,v_{n-1})$ δύο n – tuples του V_n . Ορίζουμε το εσωτερικό γινόμενο των **u** και **v** :

$\mathbf{u} \cdot \mathbf{v} = u_0 \cdot v_0 + u_1 \cdot v_1 + \ldots + u_{n-1} \cdot v_{n-1}$ όπου τα $u_i \cdot v_i$, $u_i \cdot v_i + u_{i+1} \cdot v_{i+1}$ υπολογίζονται με 2-modulo πολλαπλασιασμό και πρόσθεση αντίστοιχα . Επομένως το u·v είναι ένας βαθμωτός στο GF(2). Αν $\mathbf{u} \cdot \mathbf{v} = 0$ τότε τα **u,v** είναι ορθογώνια το ένα σε σχέση με το άλλο. Το εσωτερικό γινόμενο έχει τις εξής ιδιότητες: I) $\mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{u}$

II) $\mathbf{u} \cdot (\mathbf{v} + \mathbf{w}) = \mathbf{u} \cdot \mathbf{v} + \mathbf{u} \cdot \mathbf{w}$

III) $(a\mathbf{u}) \cdot \mathbf{v} = a(\mathbf{u} \cdot \mathbf{v})$

Έστω S είναι ένας υποχώρος του V_n διάστασης κ και έστω S_d είναι το σύνολο των διανυσμάτων του V_n τέτοιο ώστε για κάθε **u** που ανήκει στο S και **v** που ανήκει στο S_d , $u \cdot v = 0$. Το σύνολο S_d περιέχει όλα τα μηδενικά n- tuple **0**=(0,0,…,0) αφού για κάθε **u** του S $\mathbf{u} \cdot 0 = 0$. Για κάθε στοιχείο α του GF(2) και κάθε **v** που ανήκει στο S_d ,

$a \cdot \mathbf{v} = 0$ αν $a = 0$ ή $a \cdot \mathbf{v} = \mathbf{v}$ αν $a = 1$

Επομένως το $a \cdot \mathbf{v}$ ανήκει επίσης στο S_d . Έστω v και w δύο διανύσματα του S_d . Για κάθε διάνυσμα **u** του S $\mathbf{u} \cdot (v + w) = \mathbf{u} \cdot v + \mathbf{u} \cdot w = 0+0 = 0$. Από αυτό συμπεραίνουμε ότι αν τα v και w είναι ορθογώνια στο **u** τότε και το άθροισμα τους θα είναι επίσης ορθογώνιο με το **u**. Επομένως το άθροισμα v + w είναι διάνυσμα του χώρου S_d . Είναι προφανές ότι το S_d είναι ένα υπόχωρος του

V_n . Αυτός ο υπόχωρος ονομάζεται δυαδικός (ή κενός) χώρος του S και ισχύει και το αντίστροφο. Η διάσταση του S_d δίνεται από το παρακάτω θεώρημα.

Θεώρημα Έστω S είναι ένας υποχώρος διάστασης κ του διανυσματικού χώρου V_n όλων των

n-tuples ορισμένα στο GF(2). Η διάσταση του δυαδικού του S_d χώρου είναι n-k. Με άλλα λόγια ισχύει: $\dim(S) + \dim(S_d) = n$.

ΚΕΦΑΛΑΙΟ 3 - ΓΡΑΜΜΙΚΟΙ ΚΩΔΙΚΕΣ ΟΜΑΔΑΣ

Στο κεφάλαιο αυτό γίνεται μια παρουσίαση των κωδίκων ομάδας και ιδιαίτερα των γραμμικών οι οποίοι βρίσκουν καθημερινή εφαρμογή στους ψηφιακούς υπολογιστές και στα ψηφιακά δεδομένα των τηλεπικοινωνιακών συστημάτων. Δεδομένου ότι η πληροφορία κωδικοποιείται στα δυαδικά ψηφία '0' και '1' θα περιοριστούμε στους γραμμικούς κώδικες ομάδας των οποίων τα σύμβολά ορίζονται στο δυαδικό πεδίο GF(2). Η θεωρία που αναπτύσσεται για τους δυαδικούς κώδικές μπορεί να γενικευτεί σε κώδικες με σύμβολα από μη δυαδικά πεδία χωρίς κανένα περιορισμό. Οι κώδικές αυτοί αποτελούν ένα ισχυρό εργαλείο όσον αφορά την διόρθωση λαθών στην μεταδιδόμενη πληροφορία στα τηλεπικοινωνιακά συστήματα.

3.1 Εισαγωγή στους γραμμικούς κώδικες ομάδας

Θεωρούμε ότι η έξοδος μιας πηγής πληροφορίας είναι μια ακολουθία δυαδικών ψηφίων ''0'' και ''1''. Στην κωδικοποίηση ομάδας η δυαδική πληροφορία χωρίζεται σε ομάδες μηνυμάτων (message block) καθορισμένου μήκους, δηλαδή κάθε ομάδα μηνύματος, που συμβολίζεται με u, αποτελείται από κ ψηφία πληροφορίας. Υπάρχουν συνολικά 2^{κ} διακριτά μηνύματα. Ο κωδικοποιητής ,βασισμένος σε καθορισμένους κανόνες , μετατρέπει κάθε μήνυμα εισόδου u σε ένα δυαδικό n – tuple v με n > k. Αυτό το n – tuple v αναφέρεται ως **κωδική λέξη** (code word ή code vector) του μηνύματος u. Επομένως στα 2^{κ} διακριτά πιθανά μηνύματα αντιστοιχούν 2^{κ} κωδικές λέξεις. Το σύνολο των κωδικών λέξεων ονομάζεται **κώδικας ομάδας**. Για να μπορεί ο κώδικας ομάδας να είναι χρήσιμος θα πρέπει οι 2^{κ} κωδικές λέξεις να είναι διακριτές. Πρέπει δηλαδή να υπάρχει μία προς μία αντιστοιχία μεταξύ ενός μηνύματος u και της κωδικής του λέξης v.

Για ένα κώδικα ομάδας με 2^{κ} κωδικές λέξεις και μήκους n ,εκτός και αν είχε μια ειδική δομή, ο μηχανισμός κωδικοποίησης θα ήταν φοβερά πολύπλοκος για μεγάλο κ και n καθώς θα είχε να αποθηκεύσει τις 2^{κ} κωδικές λέξεις μήκους n σε ένα 'λεξικό'. Επομένως θα πρέπει να περιορίσουμε το ενδιαφέρον μας σε κώδικες ομάδας που μπορούν να είναι πρακτικοί όσον αφορά το μέγεθος τους. Είναι επίσης επιθυμητό ο κώδικας ομάδας να είναι γραμμικός γιατί με αυτόν τον τρόπο η πολυπλοκότητα κωδικοποίησης μειώνεται αισθητά.

Ορισμός : Ένας κώδικας ομάδας μήκους n και με 2^{κ} κωδικές λέξεις ονομάζεται γραμμικός (n , κ) κώδικας αν και μόνο αν οι 2^{κ} κωδικές λέξεις διαμορφώνουν ένα κ – διάστατο υπόχωρο του διανυσματικού χώρου όλων των n – tuples ορισμένα στο πεδίο GF(2).

Στην πραγματικότητα ένας δυαδικός κώδικας ομάδας είναι γραμμικός αν και μόνο αν το 2- modulo άθροισμα δύο κωδικών λέξεων είναι επίσης κωδική λέξη. Ο κώδικας ομάδας που δίνεται στον πίνακα 3.1 είναι ένας (7,4) γραμμικός κώδικας. Μπορούμε εύκολα να διαπιστώσουμε ότι το άθροισμα δύο οποιοδήποτε κωδικών λέξεων αυτού του κώδικα είναι επίσης κωδική λέξη.

Πίνακας 3.1 Γραμμικός κώδικας ομάδας με κ = 4 και n = 7

Μύνημα	Κωδική λέξη
(0 0 0 0)	(0 0 0 0 0 0 0)
(1 0 0 0)	(1 1 0 1 0 0 0)
(0 1 0 0)	(0 1 1 0 1 0 0)
(1 1 0 0)	(1 0 1 1 1 0 0)
(0 0 1 0)	(1 1 1 0 0 1 0)
(1 0 1 0)	(0 0 1 1 0 1 0)
(0 1 1 0)	(1 0 0 0 1 1 0)
(1 1 1 0)	(0 1 0 1 1 1 0)
(0 0 0 1)	(1 0 1 0 0 0 1)
(1 0 0 1)	(0 1 1 1 0 0 1)
(0 1 0 1)	(1 1 0 0 1 0 1)
(1 1 0 1)	(0 0 0 1 1 0 1)
(0 0 1 1)	(0 1 0 0 0 1 1)
(1 0 1 1)	(1 0 0 1 0 1 1)
(0 1 1 1)	(0 0 1 0 1 1 1)
(1 1 1 1)	(1 1 1 1 1 1 1)

Από τη στιγμή που ο γραμμικός (n , κ) κώδικας C είναι ένας κ – διάστατος υπόχωρος του διανυσματικού χώρου Vn όλων των δυαδικών n – tuples ,είναι πιθανό να βρεθούν κ γραμμικώς ανεξάρτητες κωδικές λέξεις , $g_0,g_1,\ldots,g_{k-1}$ του C τέτοιες ώστε κάθε κωδική λέξη v του C να είναι ένας γραμμικός συνδυασμός αυτών των κ κωδικών λέξεων ,δηλαδή :

$$v = u_0g_0 + u_1g_1+\ldots+ u_{k-1}\, g_{k-1} \quad \text{με} \quad u_i =0 \text{ ή } 1 \text{ για } 0 < i < k.$$

Αυτές τις κ γραμμικά ανεξάρτητες κωδικές λέξεις μπορούμε να τις γράψουμε ως γραμμές ενός $\kappa \times n$ πίνακα δηλαδή

$$G=\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ . \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & g_{02} & . & . & g_{0,n-1} \\ g_{10} & g_{11} & g_{12} & . & . & g_{1,n-1} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & . & . & g_{k-1,n-1} \end{bmatrix}$$

όπου $g_i= (g_{i0},g_{i1},...,g_{i,n-1})$ για $0 \le i < k$. Αν το $u = (u_0,u_1,\ldots,u_{k-1})$ είναι το μήνυμα που πρόκειται να κωδικοποιηθεί, τότε η αντίστοιχη κωδική λέξη δίνεται ως :

$v = u \cdot \mathbf{G} = (u_0,u_1,\ldots,u_{k-1}) \cdot \mathbf{G} = u_0g_0+u_1g_1+\ldots+u_{k-1}g_{k-1}$

Βλέπουμε λοιπόν ότι οι γραμμές του πίνακα **G** παράγουν τον (n,k) γραμμικό κώδικα C. Για αυτό το λόγο ο πίνακας **G** ονομάζεται **γεννήτορας πίνακας** (generator matrix) του C. Σημαντική σημείωση είναι ότι κάθε κ γραμμικά ανεξάρτητες κωδικές λέξεις ενός (n,k) γραμμικού κώδικα μπορούν να χρησιμοποιηθούν για να διαμορφώσουν το γεννήτορα πίνακα του κώδικα. Επομένως ο κωδικοποιητής έχει να αποθηκεύσει μόνο κ γραμμές του **G** και να διαμορφώσει ένα γραμμικό συνδυασμό αυτών των κ γραμμών βασισμένος στο μήνυμα εισόδου $u = (u_0,u_1,\ldots,u_{k-1})$.

Μια επιθυμητή ιδιότητα ενός γραμμικού κώδικα είναι το να έχει συστηματοποιημένη δομή των κωδικών λέξεων , δηλαδή η κωδική λέξη θα πρέπει να διαιρείται σε δύο μέρη , στο μέρος που

περιέχει το μήνυμα και στο μέρος πλεονασμού (που είναι για τον έλεγχο). Το μέρος του μηνύματος αποτελείται από κ αναλλοίωτα ψηφία πληροφορίας και το μέρος πλεονασμού ,υπεύθυνο για τον έλεγχο, αποτελείται από n – k (parity – check) ψηφία ελέγχου ισοτιμίας τα οποία είναι γραμμικά αθροίσματα των ψηφίων πληροφορίας. Ένας γραμμικός κώδικας με αυτή τη μορφή αναφέρεται ως γραμμικός συστηματικός κώδικας ομάδας. Άρα για ένα γραμμικό συστηματικό κώδικα ομάδας (7,4) τα τέσσερα δεξιά ψηφία αντιστοιχούν στα ψηφία πληροφορίας .

Τμήμα πλεονασμού για έλεγχο(n – k ψηφία)	Τμήμα πληροφορίας (κ ψηφία)

Σχήμα 3.1 Συστηματική μορφή μιας κωδικής λέξης

Ένας γραμμικός συστηματικός (n , k) κώδικας ορίζεται πλήρως από ένα $\kappa \times n$ πίνακα **G** της παρακάτω μορφής :

$$G=\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ . \\ g_{k-1} \end{bmatrix}=\begin{bmatrix} p_{00} & p_{01} & p_{02} & . & . & p_{0,n-k-1} & 1 & 0 & 0 & . & 0 \\ p_{10} & p_{11} & p_{12} & . & . & p_{1,n-k-1} & 0 & 1 & 0 & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ p_{k-1,0} & p_{k-1,1} & p_{k-1,2} & . & . & p_{k-1,n-k-1} & 0 & 0 & 0 & . & 1 \end{bmatrix}$$

όπου $p_{ij} = 0$ ή 1.Έστω με $\mathbf{I_\kappa}$ συμβολίζεται ο $k \times k$ μοναδιαίος πίνακας .Τότε ο $\mathbf{G} = [\ PI_k\ ]$. Έστω το $\mathbf{u} = (u_0,u_1,\ldots,u_{k-1})$ είναι το μήνυμα που πρόκειται να κωδικοποιηθεί. Η κωδική λέξη που αντιστοιχεί στο μήνυμα αυτό θα είναι :

$$v = (v_0, v_1, \ldots, v_{n-1}) = (u_0, u_1, \ldots, u_{k-1}) \cdot \mathbf{G}$$

Από την παραπάνω σχέση και από την μορφή του πίνακα για τα συστατικά μέρη του v έχουμε:

$$v_{n-k-i} = u_i \quad \text{για} \quad 0 \le i < k \quad \text{και} \quad (3.1)$$

$$vj = u_0 p_{0j} + u_1 p_{1j} + \ldots + u_{k-1} p_{k-1,j} \quad \text{για} \quad 0 \le j < n-k \quad (3.2)$$

Οι παραπάνω εξισώσεις δείχνουν ότι τα κ δεξιά ψηφία μιας κωδικής λέξης v αντιστοιχούν στα ψηφία πληροφορίας $u_0, u_1, \ldots, u_{k-1}$ που πρόκειται να κωδικοποιηθούν και τα υπόλοιπα n – k ψηφία πλεονασμού είναι γραμμικά αθροίσματα των ψηφίων πληροφορίας. Οι n – k εξισώσεις που δίνονται παραπάνω για τα ψηφία πλεονασμού ονομάζονται **εξισώσεις ελέγχου ισοτιμίας** (parity-check equations) του κώδικα.

Υπάρχει και ένας άλλος πολύ χρήσιμος πίνακας ο οποίος σχετίζεται με κάθε γραμμικό κώδικα ομάδας . Για κάθε $\kappa \times n$ πίνακα **G** με κ γραμμικά ανεξάρτητες γραμμές ,υπάρχει ένας

$n - k \times n$ πίνακας **H** με n – k γραμμικά ανεξάρτητες γραμμές τέτοιες ώστε κάθε διάνυσμα που ανήκει στο γραμμικό χώρο του **G** είναι ορθογώνιο με τις γραμμές του **H** και αντιστρόφως κάθε διάνυσμα που είναι ορθογώνιο στις γραμμές του **H** ανήκει στο χώρο που ορίζουν οι γραμμές του **G**.

Επομένως μπορούμε να περιγράψουμε ένα (n, k)γραμμικό κώδικα που παράγεται από τον πίνακα **G** με ένα εναλλακτικό τρόπο ως εξής : ένα n – tuple v είναι κωδική λέξη σε ένα κώδικα που παράγεται από τον **G** αν και μόνο αν $v \cdot \mathbf{H}^T = 0$. Ο πίνακας **H** καλείται πίνακας ελέγχου ισοτιμίας του κώδικα. Οι 2^{n-k} γραμμικοί συνδυασμοί των γραμμών του πίνακα **H** διαμορφώνουν ένα (n,

n – k) γραμμικό κώδικα C_d. Ο κώδικας αυτός αποτελεί τον κενό χώρο του (n,k) γραμμικού κώδικα C που παράγεται από τον πίνακα **G**. Ο C_d ονομάζεται δυαδικός κώδικας του C . Από όλα τα παραπάνω προκύπτει ότι ο πίνακας ελέγχου ισοτιμίας ενός γραμμικού κώδικα C είναι ο γεννήτορας πίνακας του δυαδικού του κώδικα C_d . Εάν ο γεννήτορας πίνακας ενός (n, k)γραμμικού κώδικα βρίσκεται σε συστηματική μορφή τότε ο πίνακας ελέγχου ισοτιμίας μπορεί να πάρει την ακόλουθη μορφή : $\mathbf{H} = [\, I_{n-k} \;\; P^T \,]$ όπου ο $\mathbf{P^T}$ είναι ο ανάστροφος του πίνακα **P**. Έστω h_j είναι η j_{th} γραμμή του **H**. Μπορούμε εύκολα να επαληθεύσουμε ότι το εσωτερικό γινόμενο της i γραμμής του **G** με την j γραμμή του **H** είναι μηδέν δηλαδή :

$$\mathbf{g_i} \cdot \mathbf{h_j} = p_{ij} + p_{ij} = 0 \quad \text{για} \quad 0 \le i < k \quad \text{και για } 0 \le j < n-k$$

γεγονός που υποδηλώνει ότι το γινόμενο $\mathbf{G} \cdot \mathbf{H}^{\mathbf{T}} = \mathbf{0}$. Επίσης οι n – k γραμμές του πίνακα **H** είναι γραμμικά ανεξάρτητες γεγονός που δείχνει ότι ο **H** είναι ένας πίνακας ελέγχου ισοτιμίας του

(n ,k) γραμμικού κώδικα που παράγεται από τον πίνακα **G** .

Οι εξισώσεις ελέγχου ισοτιμίας που δίνονται από τη σχέση (3.2) μπορούν να ληφθούν και από τον πίνακα ελέγχου ισοτιμίας **H**. Έστω $u = (u_0,u_1,\ldots,u_{k-1})$ το μήνυμα που πρόκειται να κωδικοποιηθεί. Σε συστηματική μορφή η αντίστοιχη κωδική λέξη θα είναι της μορφής

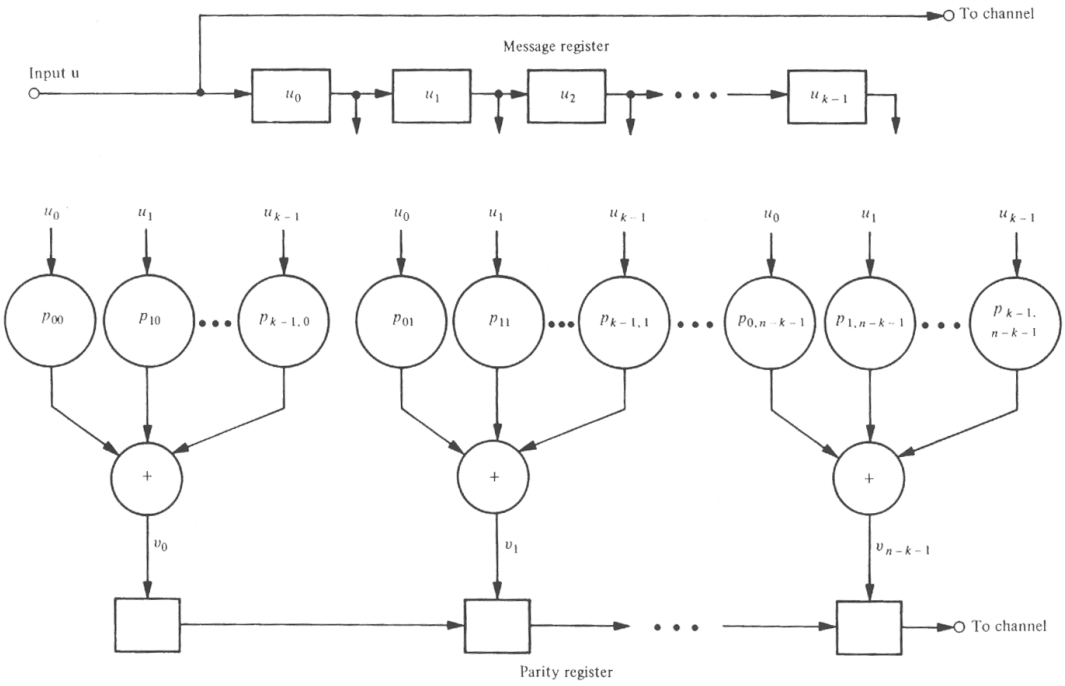
$$v = (v_0,v_1,\ldots,v_{n-k-1}, u_0,u_1,\ldots,u_{k-1})$$

Με τη χρήση της σχέσης $v \cdot \mathbf{H}^T=0$ παίρνουμε την παρακάτω σχέση

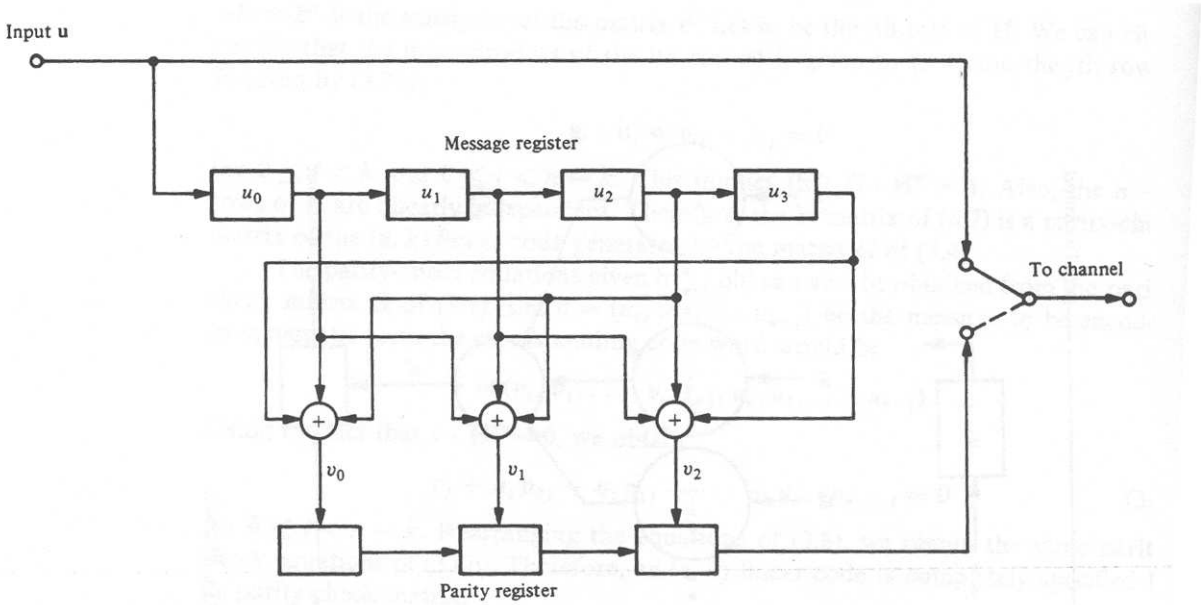
$$vj + u_0 p_{0j} + u_1 p_{1j} + \ldots + u_{k-1} p_{k-1,j} = 0 \quad \text{για} \quad 0 \le j < n-k$$

που είναι οι ίδιες εξισώσεις ελέγχου ισοτιμίας με αυτές που προκύπτουν από τη σχέση (3.2).

Βασισμένοι στις εξισώσεις των σχέσεων 3.1 και 3.2 το κύκλωμα κωδικοποίησης για έναν (n , k) γραμμικό συστηματικό κώδικα μπορεί εύκολα να υλοποιηθεί. Το κύκλωμα κωδικοποίησης φαίνεται στο παρακάτω σχήμα όπου το → δηλώνει κατάσταση ενός καταχωρητή ολίσθησης (π. χ ένα flip-flop), (+) δηλώνει ένα 2-modulo αθροιστή και (p_{ii}) δηλώνει σύνδεση μόνο αν το $p_{ij} = 1$.Η διαδικασία κωδικοποίησης είναι πολύ απλή . Το μήνυμα $u = (u_0,u_1,\ldots,u_{k-1})$ για να κωδικοποιηθεί μεταφέρεται στον καταχωρητή μυνήματος και ταυτόχρονα στο κανάλι. Μόλις το μήνυμα ολόκληρο εισέλθει στον καταχωρητή , υπολογίζονται τα n – k ψηφία για τον έλεγχο ισοτιμίας και εμφανίζονται ως έξοδοι των n – k modulo - 2 αθροιστών. Στη συνέχεια τα ψηφία ελέγχου ισοτιμίας σειριοποιούνται και μεταφέρονται στο κανάλι . Μια σημαντική παρατήρηση είναι το γεγονός ότι η πολυπλοκότητα του κυκλώματος κωδικοποίησης είναι γραμμικά ανάλογη με το μήκους του κώδικα ομάδας.



Σχήμα 3.2 Κύκλωμα κωδικοποίησης για ένα γραμμικό συστηματικό (n,k) κώδικα



Σχήμα3.3 Κύκλωμα κωδικοποίησης για ένα (7,4) συστηματικό κώδικα .

3.2 Σύνδρομα και έλεγχος λαθών

Θεωρούμε ένα (n,k) γραμμικό κώδικα με γεννήτορα πίνακα **G** και πίνακα ελέγχου ισοτιμίας **H**. Έστω $v = (v_0,v_1,\ldots,v_{n-1})$ είναι η κωδική λέξη η οποία πρόκειται να μεταδοθεί διαμέσου ενός καναλιού με θόρυβο και έστω $r = (r_0,r_1,\ldots,r_{n-1})$ το διάνυσμα που λαμβάνεται στην έξοδο του καναλιού. Εξαιτίας του θορύβου που εμφανίζει το κανάλι το r μπορεί να είναι διαφορετικό από το v. Το διανυσματικό άθροισμα $e = r + v = (e_0,e_1,\ldots,e_{n-1})$ (3.3) είναι ένα n – tuple όπου $e_i = 1$ για

$r_i \neq v_i$ και $e_i=0$ για $r_i = v_i$.Αυτό το n – tuple ονομάζεται διάνυσμα λάθους (error pattern).

Οι άσσοι (1) του e αντιστοιχούν στα λάθη που γίνονται κατά τη μετάδοση και που προκαλούνται από το θόρυβο που έχει το κανάλι. Από τη σχέση (3.3) προκύπτει ότι το διάνυσμα v που λαμβάνεται είναι το άθροισμα της μεταδιδόμενης κωδικής λέξης και του διανύσματος λάθους δηλαδή : $r = v+e$.

Φυσικά ο δέκτης δεν γνωρίζει ούτε το v ούτε το e. Με το που λαμβάνει το r ο αποκωδικοποιητής πρέπει να αποφασίσει αν το r περιέχει λάθη μετάδοσης. Αν ανιχνευθεί παρουσία λαθών τότε ο αποκωδικοποιητής θα πρέπει να αποφασίσει αν θα ενεργήσει για την εύρεση των λαθών και την διόρθωση τους (FEC)ή αν θα ζητήσει επαναμετάδοση του v(ARQ).

Όταν λαμβάνεται το r , ο αποκωδικοποιητής υπολογίζει το ακόλουθο (n – k) – tuple :

$$s= r \cdot H^T = (s_0,s_1,\ldots,s_{n-k-1}) \quad (3.4)$$

το οποίο ονομάζεται σύνδρομο (syndrome)του r. Τότε το s=0 αν και μόνο αν το r είναι κωδική λέξη και $s \neq 0$ αν και μόνο αν το r δεν είναι κωδική λέξη γεγονός που υποδηλώνει ότι έχουν συμβεί λάθη στη μετάδοση. Υπάρχει όμως και η περίπτωση λάθη μέσα σε συγκεκριμένα διανύσματα

λάθους να μην είναι ανιχνεύσιμα (το r να περιέχει λάθη αλλά το $s= r \cdot H^T = 0$) και αυτό συμβαίνει όταν το διάνυσμα λάθους είναι πανομοιότυπο με μία μη μηδενική κωδική λέξη. Σε αυτή την περίπτωση το r είναι το άθροισμα δύο κωδικών λέξεων η οποία είναι κωδική λέξη και συνεπώς

$r \cdot H^T = 0$. Διανύσματα λάθους αυτού του τύπου είναι μη ανιχνεύσιμα διανύσματα λάθους και από τη στιγμή που υπάρχουν $2^κ$-1 μη μηδενικές κωδικές λέξεις υπάρχουν και $2^κ$-1 μη ανιχνεύσιμα διανύσματα λάθους τα οποία όταν συμβαίνουν λέμε ότι ο αποκωδικοποιητής κάνει ένα λάθος αποκωδικοποίησης.

Από τη σχέση (3.4) προκύπτει :

$s_0= r_0 + r_{n-k} p_{00}+ r_{n-k-1} p_{10} +\ldots+ r_{n-1} p_{k-1,0}$

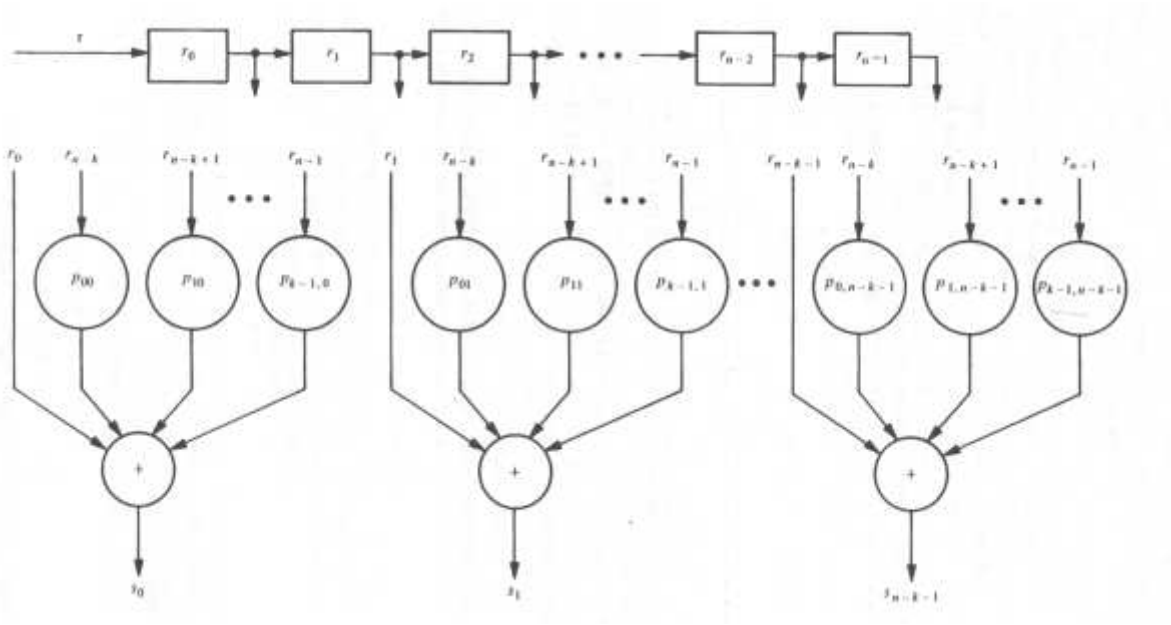
$s_1= r_1 + r_{n-k} p_{01}+ r_{n-k-1} p_{11} +\ldots+ r_{n-1} p_{k-1,1}$

.

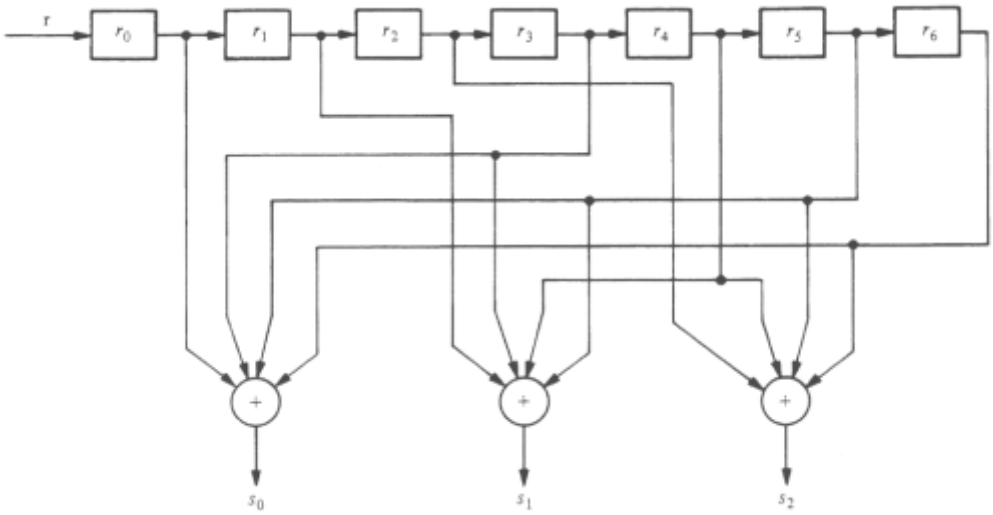
.

$s_{n-k-1}= r_{n-k-1} + r_{n-k} p_{0, n-k-1}+ r_{n-k-1} p_{1, n-k-1} +\ldots+ r_{n-1} p_{k-1, n-k-1}$

Εξετάζοντας τις παραπάνω εξισώσεις βρίσκουμε ότι το σύνδρομο s είναι απλώς το διανυσματικό άθροισμα των λαμβανόμενων ψηφίων ισοτιμίας $(r_0,r_1,\ldots,r_{n-κ-1})$ και των ψηφίων ελέγχου ισοτιμίας τα οποία έχουν υπολογιστεί ξανά από τα λαμβανόμενα ψηφία πληροφορίας $r_{n-κ}$, $r_{n-κ+1},\ldots.$, r_{n-1}. Επομένως το σύνδρομο μπορεί να διαμορφωθεί από ένα κύκλωμα παρόμοιο με αυτό του κυκλώματος κωδικοποίησης .



Σχήμα 3.4 Κύκλωμα συνδρόμου για ένα (n,k) συστηματικό κώδικα .



Στο **σχήμα 3.5** φαίνεται το κύκλωμα για το σύνδρομο για ένα (7,4) κώδικα .

Το σύνδρομο που υπολογίζεται από το λαμβανόμενο διάνυσμα r στην πραγματικότητα εξαρτάται μόνο από το διάνυσμα λάθους, και όχι από τη μεταδιδόμενη κωδική λέξη v.Από τη στιγμή που το r είναι το διανυσματικό άθροισμα των v και e τότε για το θα ισχύει :

$s= r \cdot H^T = (v+e) \cdot H^T = v \cdot H^T + e \cdot H^T$

Αλλά το $v \cdot H^T = 0$ άρα $s = e \cdot H^T$ και αν ο πίνακας ελέγχου ισοτιμίας H είναι σε συστηματική μορφή θα έχουμε :

$s_0= e_0 + e_{n-k} p_{00}+ e_{n-k-1} p_{10} +\ldots+ e_{n-1} p_{k-1,0}$

$s_1= e_1 + e_{n-k} p_{01}+ e_{n-k-1} p_{11} +\ldots+ e_{n-1} p_{k-1,1}$

.

.

$s_{n-k-1}= e_{n-k-1} + e_{n-k} p_{0, n-k-1}+ e_{n-k-1} p_{1, n-k-1} +\ldots+ e_{n-1} p_{k-1, n-k-1}$

δηλαδή τα ψηφία του συνδρόμου είναι απλώς γραμμικοί συνδυασμοί των ψηφίων του διανύσματος λάθους και μπορούν να χρησιμοποιηθούν για την διόρθωση λαθών. Άρα το πρόβλημα διόρθωσης λαθών ανάγεται ουσιαστικά στην επίλυση των n – k γραμμικών εξισώσεων για την εύρεση των λάθος ψηφίων. Γιατί όταν βρεθεί το διάνυσμα λάθους τότε το διάνυσμα r + e είναι η πραγματική μεταδιδόμενη κωδική λέξη . Δυστυχώς όμως το να βρεθεί το πραγματικό διάνυσμα λάθους δεν είναι και τόσο εύκολη υπόθεση και αυτό γιατί οι n – k γραμμικές εξισώσεις δεν έχουν μοναδική λύση αλλά έχουν 2^{κ} λύσεις. Δηλαδή υπάρχουν 2^{κ} διανύσματα λάθους που έχουν το ίδιο σύνδρομο και το πραγματικό διάνυσμα λάθους είναι ένα από αυτά. Στο σημείο αυτό ο αποκωδικοποιητής έχει να αποφασίσει από ένα σύνολο 2^{κ} υποψήφιων διανυσμάτων λάθους . Για να ελαχιστοποιηθεί η πιθανότητα μιας λανθασμένης αποκωδικοποίησης , το πιο πιθανό διάνυσμα λάθους είναι αυτό που επιλέγεται ως το σωστό και **αν το κανάλι είναι BSC τότε το πιο πιθανό διάνυσμα λάθους είναι αυτό που έχει το μικρότερο αριθμό μη μηδενικών ψηφίων.**

3.3 Η ελάχιστη απόσταση (minimum distance) ενός κώδικα ομάδας

Μια πολύ σημαντική παράμετρος ενός κώδικα ομάδας είναι η ελάχιστη απόσταση (minimum distance). Αυτή η παράμετρος καθορίζει ικανότητες του κώδικα όσον αφορά τη τυχαία ανίχνευση λαθών και τη τυχαία διόρθωση λαθών. Έστω $v = (v_0,v_1,\ldots,v_{n-1})$ ένα δυαδικό n – tuple . Το βάρος Hamming (Hamming weight) του v, που συμβολίζεται με w(v), ορίζεται ως ο αριθμός των μη μηδενικών παραγόντων του v. Για παράδειγμα το βάρος Hamming του v = (1001011) είναι 4. Έστω

v , w είναι δύο n – tuples . Η απόσταση Hamming μεταξύ των v και w που συμβολίζεται με d(v,w) ορίζεται ως ο αριθμός των θέσεων στις οποίες διαφέρουν. Για παράδειγμα η απόσταση Hamming μεταξύ των v = (1001011) και w = (0100011) είναι 3 αφού διαφέρουν στην μηδενική, πρώτη και τρίτη θέση . Η απόσταση Hamming είναι μια μετρική συνάρτηση η οποία ικανοποιεί την τριγωνική ανισότητα .Έστω v , w και x τρία n – tuples. Τότε :

$$d(v,w) + d(w,x) \geq d(v,x)$$

Από τον ορισμό της απόστασης Hamming και του ορισμού της 2-modulo πρόσθεσης ότι η απόσταση Hamming μεταξύ δύο n – tuples v και w ισούται με το βάρος Hamming του αθροίσματος των δύο n – tuples v και w δηλαδή :

$$d(v ,w) = w (v + w)$$

Έτσι δεδομένου ενός κώδικα ομάδας C μπορεί να υπολογιστεί εύκολα η απόσταση Hamming μεταξύ δύο οποιονδήποτε διακριτών κωδικών λέξεων. Η ελάχιστη απόσταση Hamming (minimum distance)του κώδικα C που συμβολίζεται με d_{min} ορίζεται ως :

$$d_{min} = \min \{ d(v,w) : v ,w \varepsilon C, v \neq w \}$$

Αν ο C είναι ένας γραμμικός κώδικας ομάδας τότε το άθροισμα δύο διανυσμάτων του είναι επίσης ένα διάνυσμα του κώδικα άρα η απόσταση Hamming μεταξύ δύο διανυσμάτων του κώδικα C είναι ίση με το βάρος Hamming ενός τρίτου διανύσματος του κώδικα C.

$$d_{min} = \min \{ w (v + w): v ,w \varepsilon C, v \neq w \}$$

$$= \min \{ w (x): x \varepsilon C, x \neq 0 \}$$

$$= w_{min}$$

Η παράμετρος w_{min} ονομάζεται ελάχιστο βάρος του γραμμικού κώδικα C. Έτσι οδηγούμαστε στο παρακάτω θεώρημα σύμφωνα με το οποίο :

Θεώρημα 1 . Η ελάχιστη απόσταση ενός γραμμικού κώδικα είναι ίση με το ελάχιστο βάρος των μη μηδενικών κωδικών λέξεων.

Επομένως προκειμένου να υπολογίσουμε την ελάχιστη απόσταση ενός γραμμικού κώδικα αρκεί να υπολογίσουμε το ελάχιστο βάρος.

Ένα επίσης βασικό θεώρημα που σχετίζει το βάρος ενός γραμμικού κώδικα ομάδας με τον πίνακα ελέγχου ισοτιμίας του κώδικα είναι το εξής :

Θεώρημα 2 . Έστω C ένας (n,k) γραμμικός κώδικας με πίνακα ελέγχου ισοτιμίας **H** . Για κάθε διάνυσμα του κώδικα με βάρος Hamming m, υπάρχουν m στήλες του **H** τέτοιες ώστε το διανυσματικό άθροισμα των m στηλών ισούται με το μηδενικό διάνυσμα. Αντιστρόφως αν υπάρχουν m στήλες του **H** των οποίων το διανυσματικό άθροισμα ισούται με το μηδενικό διάνυσμα τότε υπάρχει ένα διάνυσμα στον κώδικα που έχει βάρος Hamming m .

Από τα παραπάνω θεωρήματα προκύπτουν τα εξής πορίσματα :

Πόρισμα 1 Έστω C ένας γραμμικός κώδικας ομάδας με πίνακα ελέγχου ισοτιμίας **H**. Αν δεν υπάρχουν d – 1 οι λιγότερες στήλες του **H** που όταν προστίθενται δίνουν το μηδενικό διάνυσμα τότε ο κώδικας έχει ελάχιστο βάρος τουλάχιστον d.

Πόρισμα 2 Έστω C ένας γραμμικός κώδικας ομάδας με πίνακα ελέγχου ισοτιμίας **H**. Το ελάχιστο βάρος του C είναι ίσο με το μικρότερο αριθμό των στηλών του H που όταν προστεθούν μας δίνουν το μηδενικό διάνυσμα.

Τα παραπάνω πορίσματα χρησιμοποιούνται γενικά για να καθορίσουν ή να δώσουν ένα κατώτατο όριο στην τιμή της ελάχιστης απόστασης ενός γραμμικού κώδικα ομάδας.

3.4 Ανίχνευση λαθών και διόρθωση λαθών σε ένα κώδικα ομάδας

Όταν ένα διάνυσμα v ενός κώδικα μεταδίδεται διαμέσου ενός καναλιού με θόρυβο τότε ένα διάνυσμα λάθους m λαθών θα συμβεί και το αποτέλεσμα θα είναι το διάνυσμα r που θα λάβει ο δέκτης θα διαφέρει από το μεταδιδόμενο διάνυσμα v σε m θέσεις. Αν η ελάχιστη απόσταση σε ένα κώδικα ομάδας C είναι d_{min} τότε οποιαδήποτε δύο διακριτά διανύσματα του κώδικα διαφέρουν τουλάχιστον σε d_{min} θέσεις . Για αυτό τον κώδικά C δεν υπάρχει κανένα διάνυσμα λάθους $d_{min} - 1$ ή λιγότερων λαθών που μπορεί να αλλάξει ένα διάνυσμα του κώδικα σε ένα άλλο με αποτέλεσμα κάθε error pattern των $d_{min} - 1$ ή λιγότερων λαθών θα δώσει ένα διάνυσμα που δεν υπάρχει στο C. Μόλις ο δέκτης ανιχνεύσει ότι το διάνυσμα που έλαβε δεν είναι διάνυσμα του κώδικα C τότε ανιχνεύει ένα λάθος της μετάδοσης. Επομένως ένας κώδικας ομάδας με ελάχιστη απόσταση d_{min} είναι ικανός να ανιχνεύει όλα τα error patterns των d_{min} ή λιγότερων λαθών. Αν συμβούν d_{min} λάθη τότε υπάρχει πρόβλημα γιατί υπάρχουν δύο τουλάχιστον κωδικές λέξεις που διαφέρουν σε d_{min} θέσεις. Για αυτό το λόγο αναφέρουμε ότι η ικανότητα της τυχαίας ανίχνευσης λαθών (random-error-detection) ενός κώδικα ομάδας με ελάχιστη απόσταση d_{min} είναι $d_{min} - 1$.

Όμως ένας γραμμικός κώδικας ομάδας με ελάχιστη απόσταση d_{min} μπορεί να ανιχνεύσει ένα μεγάλο fraction of error patterns με d_{min} ή περισσότερα λάθη. Στην πραγματικότητα κάθε γραμμικός κώδικας μπορεί να ανιχνεύσει $2^n - 2^k$ διανύσματα λάθους μήκους n .Αυτό αποδεικνύεται ως εξής : Μεταξύ των $2^n - 1$ πιθανών μη μηδενικών διανυσμάτων λάθους υπάρχουν $2^k - 1$ διανύσματα λάθους που είναι πανομοιότυπα με $2^k - 1$ μη μηδενικές κωδικές λέξεις. Αν κάποιο από τα $2^k - 1$ error patterns συμβεί τότε η μεταδιδόμενη κωδική λέξη v μετατρέπεται σε μια άλλη κωδική λέξη w η οποία θα θεωρηθεί ως η μεταδιδόμενη λέξη αφού και το σύνδρομο της θα είναι μηδέν και έτσι ο δέκτης θα κάνει λανθασμένη αποκωδικοποίηση. Άρα υπάρχουν $2^k - 1$ μη ανιχνεύσιμα error patterns. Αν το διάνυσμα λάθους δεν είναι πανομοιότυπο με μία μη μηδενική κωδική λέξη, τότε το διάνυσμα που έλαβε ο δέκτης δεν θα είναι μια κωδική λέξη και το σύνδρομο του δεν θα είναι μηδέν και σε αυτή την περίπτωση το λάθος ανιχνεύεται. Υπάρχουν $2^n - 2^k$ διανύσματα λάθους που δεν αντιστοιχούν σε καμία κωδική λέξη σε ένα (n,k) γραμμικό κώδικα και τα οποία είναι προβλέψιμα διανύσματα λάθους. Για πολύ μεγάλο n το $2^k - 1$είναι μικρό σχετικά με το $2^n - 1$ επομένως λίγα διανύσματα λάθους περνάνε από το δέκτη χωρίς να ανιχνευθούν.

Έστω C ένας (n,k) γραμμικός κώδικας. Ορίζουμε ως A_i τον αριθμό των κωδικών διανυσμάτων που έχουν βάρος i στο C. Οι αριθμοί $A_0, A_1, ..., A_n$ ονομάζονται κατανομή του βάρους στο C.Αν ο C χρησιμοποιείται μόνο για την ανίχνευση λαθών σε ένα BSC (binary symmetric channel)τότε η

πιθανότητα ο αποκωδικοποιητής να αποτύχει να ανιχνεύσει την παρουσία λαθών μπορεί να υπολογιστεί από την κατανομή βάρους του C. Έστω $P_u(E)$ η πιθανότητα ενός μη ανιχνεύσιμου λάθους. Επειδή τα μη ανιχνεύσιμα λάθη συμβαίνουν μόνο όταν το διάνυσμα λάθους είναι πανομοιότυπο με ένα μη μηδενικό διάνυσμα του C θα έχουμε :

$$P_u(E) = \sum_{i=1}^{n} A_i p^i (1 - p)^{n-1}$$

Όπου ρ είναι η πιθανότητα μετάδοσης σε ένα BSC.Αν η ελάχιστη απόσταση του C είναι d_{min} τότε από το A_1 μέχρι το A_{dmin-1} είναι μηδέν.

Για να βρούμε πόσα λάθη μπορεί να διορθώσει ένας κώδικας ομάδας C με ελάχιστη απόσταση d_{min} εργαζόμαστε ως εξής : Η ελάχιστη απόσταση d_{min} μπορεί να είναι είτε περιττός είτε άρτιος αριθμός. Έστω t είναι ένας θετικός ακέραιος αριθμός τέτοιος ώστε :

$$2t + 1 \leq d_{min} \leq 2t + 2$$

Θα αποδειχτεί ότι ο κώδικας C μπορεί να διορθώνει μέχρι t ή λιγότερα λάθη. Έστω v και r είναι το μεταδιδόμενο κωδικό διάνυσμα και το διάνυσμα που λαμβάνεται στο δέκτη αντίστοιχα και w ένα οποιοδήποτε διάνυσμα του κώδικα C. Οι αποστάσεις Hamming μεταξύ των διανυσμάτων v, r και w ικανοποιούν την τριγωνική ανισότητα :

$$d(v,r) + d(w,r) \geq d(v,w) \quad (a)$$

Υποθέτουμε ότι ένα κατά τη διάρκεια της μετάδοσης του v συμβαίνει ένα διάνυσμα λάθους t' λαθών. Τότε το μήνυμα που λαμβάνεται θα διαφέρει από το v σε t' θέσεις και έτσι θα έχουμε $d(v,r)= t'$. Αφού τα v και w είναι διανύσματα του C τότε θα έχουμε :

$$d(v,w) \geq d_{min} \geq 2t+1 \quad (\beta)$$

Από τις σχέσεις (α) και (β) προκύπτει ότι : $d(w,r) \geq 2t+1 - t'$

Αν $t' \leq t$ τότε $d(w,r) > t$

Η παραπάνω ανισότητα δείχνει ότι αν συμβεί ένα διάνυσμα λάθους με t ή λιγότερα λάθη τότε το διάνυσμα που λαμβάνεται r βρίσκεται πιο κοντά (από άποψη απόστασης Hamming) στο μεταδιδόμενο κωδικό διάνυσμα v από οποιοδήποτε άλλο διάνυσμα w του κώδικα. Για ένα BSC, αυτό σημαίνει ότι η υπό συνθήκη πιθανότητα $P(r|v)$ είναι μεγαλύτερη από την υπό συνθήκη

πιθανότητα P(r |w) για $w \neq v$. Βασισμένοι στης μέγιστης πιθανότητας σχέδιο αποκωδικοποίησης, το r αποκωδικοποιείται στο v, που είναι και το πραγματικό μεταδιδόμενο διάνυσμα του κώδικα γεγονός που συνεπάγεται την διόρθωση των λαθών. Ο κώδικας όμως δεν μπορεί να διορθώνει διανύσματα λάθους m λαθών με $m > t$, από τη στιγμή που υπάρχει μία περίπτωση το διάνυσμα που λήφθηκε με m λάθη να είναι πιο κοντά σε ένα λανθασμένο διάνυσμα από ότι στο πραγματικό που μεταδόθηκε.

Έτσι συνοπτικά έχουμε ότι ένας κώδικας ομάδας με ελάχιστη απόσταση d_{min} εγγυάται την διόρθωση όλων των λανθασμένων διανυσμάτων με $t=[(d_{min} -1) /2]$ ή λιγότερων λαθών όπου το $[(d_{min} -1) /2]$ υποδηλώνει τον μεγαλύτερο ακέραιο που δεν είναι μεγαλύτερος από $(d_{min} -1) /2$. Η παράμετρος $(d_{min} -1) /2$ ονομάζεται **ικανότητα διόρθωσης τυχαίων λαθών** (random-error-correcting capability) του κώδικα. Ο κώδικας αυτός αναφέρεται και σαν **t-error-correcting code** και αν χρησιμοποιηθεί για την διόρθωση ενός BSC με πιθανότητα μετάδοσης p τότε η πιθανότητα να γίνει λάθος στη αποκωδικοποίηση είναι φραγμένη άνω :

$$P(E) \leq \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-I}$$

Πρακτικά ένας κώδικας μπορεί συχνά να διορθώνει λ ή λιγότερα λάθη και ταυτόχρονα να ανιχνεύει m(m>λ) ή λιγότερα λάθη(δηλαδή μπορεί να ανιχνεύει την παρουσία περισσότερων λαθών χωρίς να κάνει λάθος αποκωδικοποίηση).Για αυτό το λόγο η ελάχιστη απόσταση d_{min} αυτού του κώδικα είναι τουλάχιστον λ+ m+1.

3.5 Στάνταρ πίνακας (standard array) και αποκωδικοποίηση συνδρόμου

Σ' αυτήν την παράγραφο παρουσιάζεται το σχέδιο αποκωδικοποίησης για τους γραμμικούς κώδικες ομάδας. Έστω $v_1, v_2, \ldots, v_{2^k}$ τα διανύσματα ενός (n , k) κώδικα C. Ανεξάρτητα ποιο κωδικό διάνυσμα μεταδίδεται διαμέσου ενός καναλιού με θόρυβο, το λαμβανόμενο διάνυσμα r μπορεί να είναι ένα από τα 2^n n-tuples ορισμένα στο GF(2).Οποιοδήποτε σχέδιο αποκωδικοποίησης που χρησιμοποιείται στον δέκτη είναι ένας κανόνας που αντιστοιχεί τα 2^n πιθανά λαμβανόμενα διανύσματα στα 2^κ ασύνδετα υποσύνολα $D_1, D_2, \ldots, D_2^k$ τέτοια ώστε το κωδικό διάνυσμα v_i να

περιέχεται στο υποσύνολο D_i για $1 \leq i \leq 2^k$. Επομένως κάθε υποσύνολο D_i είναι ένα προς ένα αντιστοιχισμένο σε ένα κωδικό διάνυσμα v_i .Αν ένα λαμβανόμενο διάνυσμα r βρεθεί να ανήκει στο D_i τότε το r αποκωδικοποιείται στο v_i .Σωστή κωδικοποίηση λοιπόν γίνεται αν και μόνο αν το λαμβανόμενο διάνυσμα ανήκει σε ένα υποσύνολο D_i που αντιστοιχεί στο πραγματικό διάνυσμα που μεταδόθηκε.

Μια μέθοδος συμμετοχής των 2^n πιθανών λαμβανόμενων διανυσμάτων στα 2^κ ασύνδετα υποσύνολα τέτοια ώστε το καθένα από αυτά να περιλαμβάνει ένα και μόνο ένα διάνυσμα περιγράφεται σ΄ αυτό το σημείο και βασίζεται στη γραμμική δομή του κώδικα. Πρώτα από όλα τα 2^κ κωδικά διανύσματα του C τοποθετούνται σε μία γραμμή μαζί με μηδενικό διάνυσμα

$v_1 = (0,0....,0)$ που είναι το πρώτο αριστερό στοιχείο της γραμμής. Από τα εναπομείναντα $2^n - 2^\kappa$

n-tuples, ένα n-tuple e_2 επιλέγεται να τοποθετηθεί κάτω από το μηδενικό διάνυσμα v_1. Τώρα διαμορφώνεται η δεύτερη γραμμή με το να προσθέσουμε το e_2 σε κάθε κωδικό διάνυσμα v_i της πρώτης γραμμής και τοποθετώντας το άθροισμα e_2+v_i κάτω από το v_i.Έχοντας συμπληρώσει τη δεύτερη γραμμή από τα εναπομείναντα $2^n - 2^\kappa$ n-tuples, ένα αχρησιμοποίητο n-tuple e_3 επιλέγεται να τοποθετηθεί κάτω από το μηδενικό διάνυσμα v_1. Τώρα διαμορφώνεται η τρίτη γραμμή με το να προσθέσουμε το e_3 σε κάθε κωδικό διάνυσμα v_i της πρώτης γραμμής και τοποθετώντας το άθροισμα e_3+v_i κάτω από το v_i . Η διαδικασία αυτή συνεχίζεται έως ότου χρησιμοποιηθούν όλα τα n-tuples και με αυτό τον τρόπο δημιουργείται ένας πίνακας με γραμμές και στήλες όπως αυτές που φαίνονται παρακάτω και ο πίνακας αυτός ονομάζεται στάνταρ πίνακας του δοσμένου γραμμικού κώδικα C :

$v_1=0$	v_2…..	v_i…….	v_{2^k}
$\mathbf{e_2}$	$\mathbf{e_2}+v_2$…..	$\mathbf{e_2}+v_i$…….	$\mathbf{e_2}+v_{2^k}$
$\mathbf{e_3}$	$\mathbf{e_3}+v_2$…..	$\mathbf{e_3}+v_i$…….	$\mathbf{e_3}+v_{2^k}$
.			
.			
.			
$\mathbf{e_{2^{n-k}}}$	$\mathbf{e_{2^{n-k}}}+v_2$…..	$\mathbf{e_{2^{n-k}}}+v_i$…….	$\mathbf{e_{2^{n-k}}}+v_{2^k}$

Σχήμα 3.6 Στάνταρ πίνακας για ένα (n,k)γραμμικό κώδικα

Από τον κανόνα δόμησης του στάνταρ πίνακα ότι το άθροισμα δύο οποιονδήποτε διανυσμάτων της ίδιας γραμμής είναι ένα διάνυσμα του κώδικα C.

Ένα βασικό και πολύ χρήσιμο θεώρημα για το στάνταρ πίνακα είναι το εξής :

Θεώρημα Δεν υπάρχουν δύο n-tuples στην ίδια γραμμή του στάνταρ πίνακα που να είναι πανομοιότυπα. Κάθε n-tuple εμφανίζεται σε μία και μόνο μία γραμμή του πίνακα.

Από το θεώρημα αυτό προκύπτει ότι υπάρχουν $2^n/2^k = 2^{n-k}$ διαφορετικές γραμμές στο στάνταρ πίνακα και κάθε γραμμή περιέχει 2^κ διακριτά στοιχεία. Οι 2^{n-k} γραμμές ονομάζονται cosets του κώδικα C και το πρώτο n-tuple e_j κάθε coset ονομάζεται coset leader.

Ένας στάνταρ πίνακας ενός (n , k) γραμμικού κώδικα C αποτελείται από 2^κ διαφορετικές στήλες και κάθε στήλη αποτελείται από 2^{n-k} n-tuples με το στοιχείο στην κορυφή της στήλης να είναι διάνυσμα του C.Έστω με D_j συμβολίζεται η j στήλη του στάνταρ πίνακα. Τότε :

$$D_j = \{ v_j, e_2+v_j, e_3+v_j, \ldots, \mathbf{e_{2^{n-k}}}+v_j \} \quad (\gamma)$$

Όπου v_j είναι ένα διάνυσμα του κώδικα C και $e_2, e_3, \ldots, \mathbf{e_{2^{n-k}}}$ είναι τα coset leader. Οι 2^κ διαφορετικές στήλες $D_1, D_2, \ldots, D_{2^k}$ μπορούν να χρησιμοποιηθούν για την αποκωδικοποίηση του κώδικα C. Υποθέτουμε ότι το κωδικό διάνυσμα v_j μεταδίδεται διαμέσου ενός καναλιού με θόρυβο. Από τη σχέση (γ) μπορούμε να δούμε ότι το λαμβανόμενο διάνυσμα r ανήκει στο D_j αν το διάνυσμα λάθους που προκαλείται από το κανάλι είναι ένα coset leader.Όταν συμβεί το παραπάνω το λαμβανόμενο διάνυσμα r αποκωδικοποιείται σωστά στο μεταδιδόμενο κωδικό διάνυσμα v_j . Απεναντίας αν το διάνυσμα λάθους που θα προκληθεί από το κανάλι με θόρυβο δεν είναι ένα coset leader τότε θα έχουμε μια λανθασμένη αποκωδικοποίηση. Για τους προαναφερθέντες λόγους τα

2^{n-k} coset leaders ονομάζονται correctable error patterns. Συνοψίζοντας τα αποτελέσματα αυτά έχουμε το παρακάτω θεώρημα:

Θεώρημα Κάθε (n , k) γραμμικός κώδικας ομάδας είναι ικανός να διορθώσει 2^{n-k} error patterns.

Για να ελαχιστοποιηθεί η πιθανότητα μιας λανθασμένης αποκωδικοποίησης τα error patterns τα οποία είναι πιο πιθανά να συμβούν σε ένα δοσμένο κανάλι θα πρέπει να επιλέγονται για coset leaders.Για ένα BSC το διάνυσμα λάθους μικρότερου βάρους είναι πολύ πιθανότερο από ένα διάνυσμα λάθους μεγαλύτερου βάρους. Επομένως όταν φτιάχνεται ο στάνταρ πίνακας θα πρέπει κάθε coset leader να επιλέγεται έτσι ώστε να έχει το μικρότερο βάρος από τα διαθέσιμα εναπομείναντα διανύσματα. Διαλέγοντας με αυτό τον τρόπο coset leaders, κάθε coset leader θα έχει

το μικρότερο βάρος του coset στο οποίο ανήκει επομένως η αποκωδικοποίηση που βασίζεται στο στάνταρ πίνακα είναι η ελάχιστης απόστασης αποκωδικοποίησης.

Έστω **r** λαμβανόμενο διάνυσμα το οποίο υποθέτουμε ότι το βρίσκουμε στη i στήλη D_i και στο m coset του στάνταρ πίνακα. Άρα το r αποκωδικοποιείται στο κωδικό διάνυσμα v_i . Από τη στιγμή που το $r = e_m + v_i$ η απόσταση μεταξύ του r και του v_i θα είναι :

$d(r ,v_i) = w(r +v_i) = w (e_m + v_i + v_i) =w(e_m)$

Η απόσταση μεταξύ του r και οποιουδήποτε άλλου διανύσματος π.χ v_j θα είναι :

$d(r ,v_j) = w(r +v_j) = w (e_m + v_i + v_j)$

αφού τα v_i ,και v_j είναι δύο διαφορετικά διανύσματα τότε το άθροισμα τους θα ισούται με ένα μη μηδενικό διάνυσμα έστω v_s . Άρα θα έχουμε :

$d(r ,v_j) = w (e_m + v_s)$

από τη στιγμή που τα e_m , $e_m + v_s$ ανήκουν στο ίδιο coset $w (e_m) \leq w (e_m + v_s)$ και από αυτό συνεπάγεται ότι : $d(r ,v_i) \leq d(r ,v_j)$

Το συμπέρασμα είναι ότι το λαμβανόμενο διάνυσμα αποκωδικοποιείται στο πιο κοντινό κωδικό διάνυσμα. Αν λοιπόν κάθε coset leader επιλέγεται να έχει το μικρότερο βάρος στο coset του, η αποκωδικοποίηση που βασίζεται στο στάνταρ πίνακα είναι η ελάχιστης απόστασης αποκωδικοποίηση (ή MLD minimum distance decoding).

Έστω α_i ο αριθμός των coset leaders με βάρος i.Οι αριθμοί $\alpha_0,\alpha_1,\alpha_2,...,\alpha_n$ ονομάζονται κατανομή βάρους των coset leaders.Δεδομένου ότι τα λάθη αποκωδικοποίησης συμβαίνουν αν και μόνο αν τα διανύσματα λάθους δεν είναι coset leaders η πιθανότητα λάθους για ένα BSC με πιθανότητα μετάδοσης ρ θα είναι :

$$P(E) = 1 - \sum_{i=0}^{n} a_i p^i (1-p)^{n-i}$$

Ένας (n ,k) γραμμικός κώδικας είναι ικανός να ανιχνεύει $2^n - 2^k$ διανύσματα λάθους αλλά είναι σε θέση να διορθώνει μόνο τα 2^{n-k} από αυτά. Για μεγάλο n, το 2^{n-k} είναι πολύ μικρό σε σχέση με το $2^n - 2^k$.Για αυτό το λόγο η πιθανότητα να γίνει λανθασμένη αποκωδικοποίηση είναι πολύ μεγαλύτερη από την πιθανότητα ενός μη ανιχνεύσιμου λάθους .

Θεώρημα Για ένα (n ,k) γραμμικό κώδικα C με ελάχιστη απόσταση d_{min} όλα τα n- tuples που έχουν βάρος t μικρότερου ή ίσου με $[(d_{min} -1)/2]$ μπορούν να χρησιμοποιηθούν ως coset leaders ενός στάνταρ πίνακα του C. Αν όλα τα n- tuples βάρους μικρότερου ή ίσου t χρησιμοποιηθούν ως coset leaders, τότε υπάρχει τουλάχιστον ένα n- tuple βάρους t + 1 το οποίο δεν μπορεί να χρησιμοποιηθεί ως coset leader.

Έτσι βλέπουμε ότι ο στάνταρ πίνακας έχει την σημαντική ιδιότητα ότι μπορεί να χρησιμοποιηθεί για την απλοποίηση της διαδικασίας αποκωδικοποίησης. Θεωρούμε τώρα τον πίνακα **H** ως πίνακα ελέγχου ισοτιμίας ενός δεδομένου (n , k) γραμμικού κώδικα C.

Θεώρημα Όλα τα 2^{κ} n-tuples ενός coset leader έχουν το ίδιο σύνδρομο. Τα σύνδρομα για διαφορετικά cosets είναι διαφορετικά.

Υπενθυμίζουμε ότι το σύνδρομο ενός n-tuple είναι ένα (n – k)- tuple και υπάρχουν 2^{n-k} διακριτά

(n – k)- tuples.Από το τελευταίο θεώρημα προκύπτει ότι υπάρχει μία προς μία αντιστοιχία μεταξύ ενός coset leader και ενός συνδρόμου. Χρησιμοποιώντας αυτή τη σχέση μπορούμε να διαμορφώσουμε τον πίνακα αποκωδικοποίησης που είναι πιο απλός από τον στάνταρ πίνακα. Ο πίνακας αυτός περιέχει 2^{n-k} coset leaders και τα αντίστοιχα σύνδρομα τους. Αυτός ο πίνακας είτε αποθηκεύεται είτε στέλνεται (wired)στον δέκτη.

Η αποκωδικοποίηση ενός λαμβανόμενου διανύσματος περιλαμβάνει τρία βήματα :

- Βήμα 1.Υπολογίζεται το σύνδρομο του r, $r \cdot H^T$.
- Βήμα 2.Τοποθετείται το coset leader e_l που το σύνδρομο του είναι ίσο με το $r \cdot H^T$.Τότε το e_l θεωρείται ότι είναι το διάνυσμα λάθους που προκλήθηκε από το κανάλι.
- Βήμα 3 .Αποκωδικοποιείται το λαμβανόμενο διάνυσμα στο κωδικό διάνυσμα $v = r + e_l$.

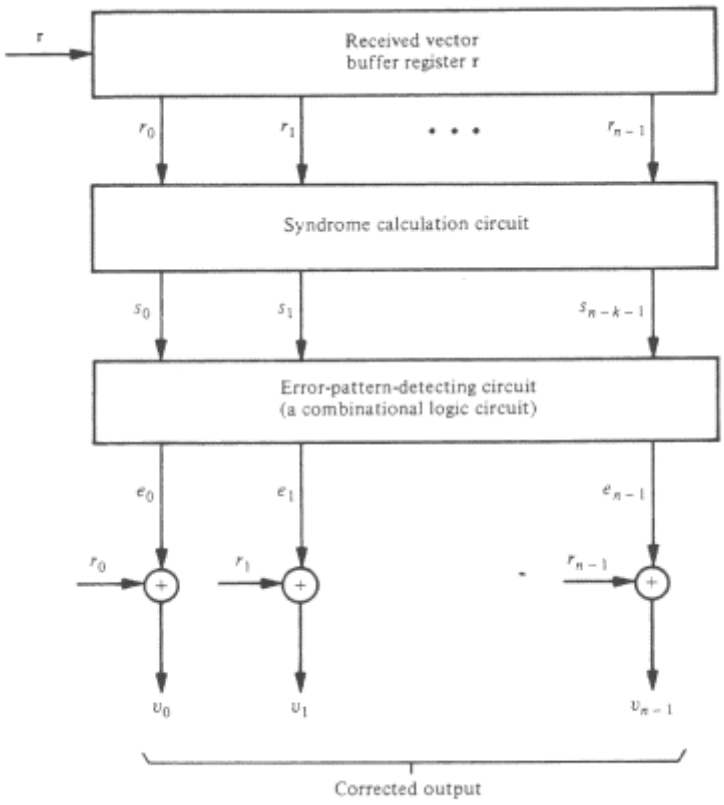
Τα τρία παραπάνω βήματα αποτελούν ένα σχέδιο αποκωδικοποίησης που ονομάζεται αποκωδικοποίηση συνδρόμου (ή table lookup decoding). Το σχέδιο αυτό οδηγεί στη μικρότερη καθυστέρηση αποκωδικοποίησης και στη μικρότερη πιθανότητα λάθους. Αλλά για μεγάλο n-k η υλοποίηση αυτού του σχεδίου γίνεται μη πρακτικό και είτε χρειάζεται μεγάλο χώρο αποθήκευσης είτε είναι περίπλοκο το λογικό κύκλωμα που χρησιμοποιείται για την υλοποίηση του. Μερικά πρακτικά σχέδια αποκωδικοποίησης τα οποία είναι παραλλαγές του table - lookup decoding παρουσιάζονται στα παρακάτω κεφάλαια. Για κάθε ένα από αυτά τα σχέδια αποκωδικοποίησης

απαιτείται να διαθέτουν κάποιες επιπρόσθετες ιδιότητες οι κώδικες για τους οποίους υλοποιούνται πέραν της γραμμικής δομής.

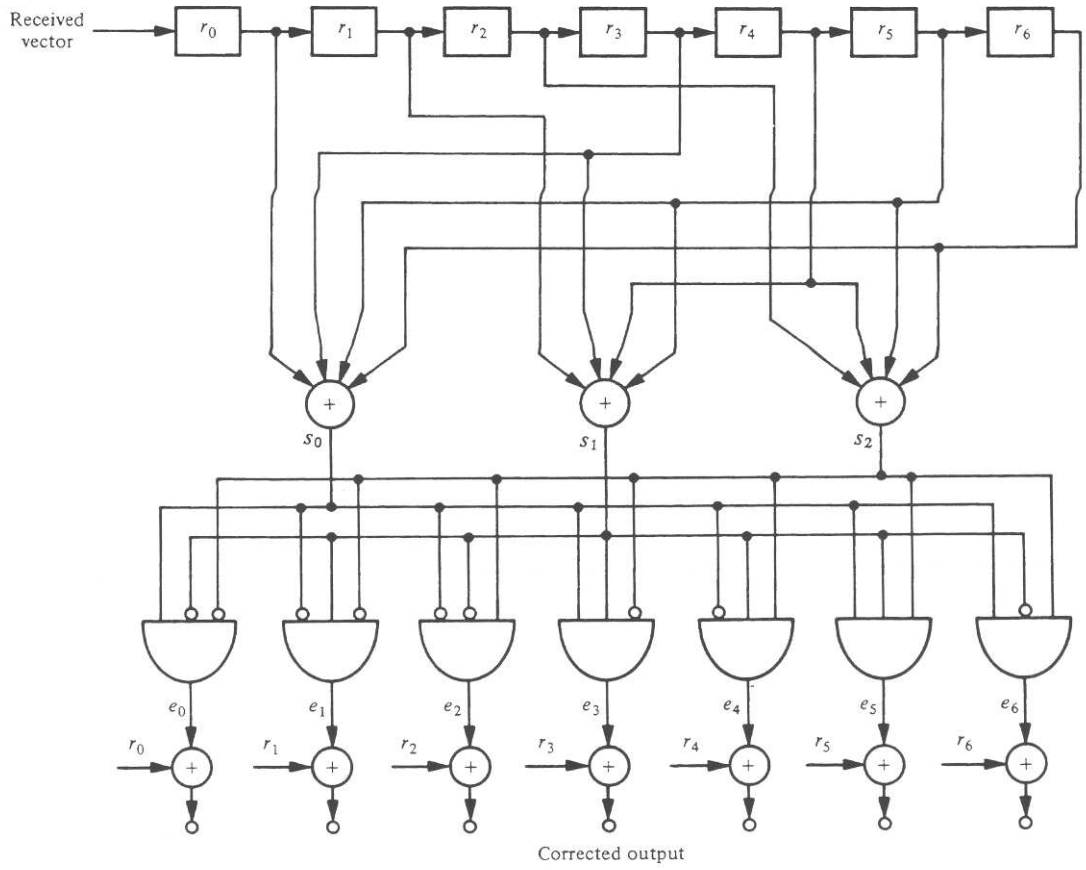
Όσον αφορά τον πίνακα αποκωδικοποίησης σε ένα table - lookup decoding μπορεί να θεωρηθεί ως ο πίνακας αληθείας n switching συναρτήσεων :

$e_0 = f_0 (s_0, s_1,\ldots s_{n-k-1})$,
$e_1 = f_1 (s_0, s_1,\ldots s_{n-k-1})$,
.
.
$e_{n-1} = f_{n-1} (s_0, s_1,\ldots s_{n-k-1})$,
όπου $s_0, s_1,\ldots s_{n-k-1}$ είναι τα ψηφία του συνδρόμου που θεωρούνται ως οι switching μεταβλητές, και τα $e_0, e_1,\ldots e_{n-1}$ είναι τα υπολογισμένα ψηφία λάθους.

Όταν αυτές οι n switching συναρτήσεις προκύψουν και απλοποιηθούν, ένα συνδυαστικό λογικό κύκλωμα με n – k ψηφία συνδρόμου για εισόδους και τα υπολογισμένα ψηφία λάθους για εξόδους μπορεί να πραγματοποιηθεί. Ο γενικός αποκωδικοποιητής για ένα (n , k) γραμμικό κώδικα βασισμένο στο σχέδιο table - lookup decoding φαίνεται στο παρακάτω σχήμα. Το κόστος αυτού του αποκωδικοποιητή εξαρτάται βασικά από την πολυπλοκότητα του συνδυαστικού λογικού κυκλώματος.



Σχήμα 3.7 Γενικός αποκωδικοποιητής για ένα γραμμικό κώδικα ομάδας



Σχήμα 3.8 Κύκλωμα αποκωδικοποίησης για ένα (7,4) κώδικα που δίνεται στον πίνακα 3.1

3.6 Πιθανότητα ενός μη ανιχνεύσιμου λάθους για ένα γραμμικό κώδικα σε ένα BSC(binary symmetric channel)

Έστω { A_0 ,$A_{1,...,}$ A_n } η κατανομή του βάρους σε ένα γραμμικό κώδικα C και { B_0 ,$B_{,...,}$ B_n } η κατανομή του βάρους του δυαδικού του κώδικα C_d . Σε μορφή πολυωνύμου οι δύο παραπάνω κατανομές είναι:

$A(z) = A_0 + A_1 z +_{...} +A_n z^n$

$B(z) = B_0 + B_1 z +_{...} +B_n z^n$

Τότε τα A(z), B(z) συνδέονται με την παρακάτω ταυτότητα :

$$A(z) = 2^{-(n-k)} (1+z)^{n} B\left(\frac{1-z}{1+z}\right)$$

Η ταυτότητα αυτή είναι γνωστή ως η Mac Williams ταυτότητα. Τα πολυώνυμα A(z), B(z) ονομάζονται (enumerators) βάρους του (n , k) γραμμικό κώδικα C και του δυαδικού του κώδικα C_d.

Για τον υπολογισμό της πιθανότητας ενός μη ανιχνεύσιμου λάθους για ένα (n , k) κώδικα έχουμε :

$$P_u(E) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i} = (1-p)^n \sum_{i=1}^{n} A_i \left(\frac{p}{1-p}\right)^i$$

Αντικαθιστώντας το $z = \frac{p}{1-p}$ στο A(z) και χρησιμοποιώντας το γεγονός ότι $A_0 = 1$ έχουμε την παρακάτω ταυτότητα :

$$A\left(\frac{p}{1-p}\right) - 1 = \sum_{i=1}^{n} A_i \left(\frac{p}{1-p}\right)^i$$

Άρα το $P_u(E) = (1-p)^n A\left(\frac{p}{1-p}\right) - 1$ (3.6^{α})

Έτσι με τη χρήση της Mac Williams ταυτότητας τελικά παίρνουμε την εξής σχέση για το $P_u(E)$:

$$P_u(E) = 2^{-(n-k)} B(1-2p) - (1-p)^n, \quad (3.6^{\beta})$$

Όπου $B(1-2p) = \sum_{i=1}^{n} B_i (1-2p)^i$.

Επομένως υπάρχουν δύο τρόποι για τον υπολογισμό της πιθανότητας ενός μη ανιχνεύσιμου λάθους για ένα γραμμικό κώδικα και συχνά ο ένας είναι πιο εύκολος από τον άλλο. Αν n – k είναι μικρότερο από το κ, είναι πιο εύκολο να υπολογιστεί το $P_u(E)$ από τη σχέση (3.6^{β}) διαφορετικά χρησιμοποιούμε τη σχέση (3.6^{α}).

Θεωρητικά μπορούμε να υπολογίσουμε την κατανομή βάρους ενός (n , k) γραμμικού κώδικα εξετάζοντας τις 2^{κ} κωδικές λέξεις ή εξετάζοντας $2^{n-\kappa}$ κωδικές λέξεις του δυαδικού του κώδικα με τη βοήθεια της ταυτότητας Mac Williams.Όμως για μεγάλα n, k ,n- k ο υπολογισμός πρακτικά

γίνεται αδύνατος. Εκτός κάποιων μικρών γραμμικών κωδίκων και μια μικρή κλάση αυτών , η κατανομή βάρους για αρκετούς γνωστούς κώδικες είναι ακόμα άγνωστοι, επομένως είναι δύσκολο αν όχι αδύνατο να υπολογιστεί η πιθανότητα ενός μη ανιχνεύσιμου λάθους.

Παρόλο όμως που είναι δύσκολο να υπολογιστεί η πιθανότητα ενός μη ανιχνεύσιμου λάθους είναι αρκετά εύκολο να υπολογιστεί ένα άνω φράγμα της μέσης πιθανότητας ενός μη ανιχνεύσιμου λάθους για το σύνολο των (n ,k) γραμμικών συστηματικών κωδίκων. Γνωρίζουμε ότι ένας (n ,k) συστηματικός κώδικας καθορίζεται πλήρως από το γεννήτορα πίνακα G . Ο υποπίνακας P αποτελείται από κ(n – k) εισόδους. Από τη στιγμή που η κάθε είσοδος p_{ij} είναι είτε 0 είτε 1 τότε υπάρχουν $2^{\kappa(n-\kappa)}$ διακριτοί γεννήτορες πίνακες G' . Έστω Γ το σύνολο των κωδίκων που παράγονται από αυτούς τους $2^{\kappa(n-\kappa)}$ γεννήτορες πίνακες. Υποθέτουμε ότι μπορούμε να διαλέξουμε τυχαία από το Γ ένα κώδικα και να τον χρησιμοποιήσουμε για ανίχνευση λαθών. Έστω C_j είναι ένας επιλεγμένος κώδικας. Η πιθανότητα του να επιλεγεί ο C_j κώδικας είναι :

$$P(C_j)= 2^{-\kappa(n-\kappa)}$$

Και έστω A_{ji} συμβολίζει τον αριθμό των κωδικών λέξεων του C_j με βάρος i.Η πιθανότητα ενός μη ανιχνεύσιμου λάθους για τον C_j δίνεται από τη σχέση :

$$P_u (E| C_j) = \sum_{i=1}^{n} A_{ji}\, p^i (1 - p)^{n-i} .$$

Η μέση πιθανότητα για ένα μη ανιχνεύσιμο λάθος ενός γραμμικού κώδικα που ανήκει στο Γ θα είναι :

$$P_u (E) = \sum_{j=1}^{\Gamma} P (C_j) P_u (E| C_j),$$

Όπου Γ ο αριθμός των κωδικών στο Γ. Με τον συνδυασμό των τριών παραπάνω σχέσεων έχουμε :

$$P_u (E) = 2^{-\kappa(n-\kappa)} \sum_{i=1}^{n} p^i (1 - p)^{n-i} \sum_{j=1}^{\Gamma} A_{ji} .$$

Ένα μη μηδενικό n – tuple είτε περιλαμβάνεται σε ακριβώς $2^{(\kappa-1)(n-\kappa)}$ κώδικες του Γ ή δεν περιλαμβάνεται σε κανένα από αυτούς. Από τη στιγμή που υπάρχουν $\binom{n}{i}$ **n –tuples** βάρους i τότε θα έχουμε :

$$\sum_{j=1}^{\Gamma} A_{ji} \leq \left(\frac{n}{i}\right) 2^{(\kappa-1)(n-\kappa)}$$

Από όλα τα παραπάνω για το άνω φράγμα της μέσης πιθανότητας ενός μη ανιχνεύσιμου λάθους για ένα (n ,k) γραμμικό συστηματικό κώδικα θα έχουμε :

$$P_u(E) \leq 2^{-(n-\kappa)} \sum_{i=1}^{n} \left(\frac{n}{i}\right) p^i (1-p)^{n-i} = 2^{-(n-\kappa)} [1-(1-p)^n]$$

Αφού το $[1-(1-p)^n] \leq 1$ είναι ξεκάθαρο ότι $P_u(E) \leq 2^{-(n-\kappa)}$

Με άλλα λόγια η πιθανότητα $P_u(E)$ είναι φραγμένη άνω από $2^{-(n-\kappa)}$ και μειώνεται εκθετικά σε σχέση με τον αριθμό των ψηφίων ισοτιμίας ελέγχου (n – k).Αυτό φυσικά δεν είναι ακόμα γνωστό αν ισχύει για όλους τους γραμμικούς κώδικες. Στην επόμενη παράγραφο παρουσιάζεται μία κλάση κωδίκων οι οποίοι ικανοποιούν το άνω φράγμα που παρουσιάστηκε για την πιθανότητα ενός μη ανιχνεύσιμου λάθους.

3.7 Κώδικες Hamming

Οι κώδικες Hamming είναι οι πρώτοι κώδικες που επινοήθηκαν για τη διόρθωση λαθών. Αυτοί οι κώδικες και οι παραλλαγές τους χρησιμοποιήθηκαν ευρέως για τη διόρθωση λαθών στις ψηφιακές τηλεπικοινωνίες και στα συστήματα αποθήκευσης δεδομένων.

Για κάθε θετικό ακέραιο αριθμό $m \geq 3$ υπάρχει ένας Hamming κώδικας με τις εξής παραμέτρους :

Μήκος κώδικα :	$n = 2^m - 1$
Αριθμός των συμβόλων πληροφορίας :	$k = 2^m - m - 1$
Αριθμός των συμβόλων ελέγχου ισοτιμίας :	$n - k = m$
Ικανότητα διόρθωσης λαθών :	$t = 1 (d_{min} = 3)$

Ο πίνακας ελέγχου ισοτιμίας Η αυτού του κώδικα περιέχει όλα τα μη μηδενικά m – tuples ως στήλες. Σε συστηματική μορφή οι στήλες του Η έχουν την ακόλουθη μορφή :

$$H = [\ I_m \quad Q\ ]$$

Όπου I_m είναι ένας m × m μοναδιαίος πίνακας και ο υποπίνακας Q αποτελείται από $2^m - m - 1$

στήλες που είναι τα m – tuples βάρους 2 ή περισσότερο. Οι στήλες του Q μπορούν να μπουν χωρίς καμιά διάταξη χωρίς να επηρεάσουν την ιδιότητα της απόστασης και την κατανομή βάρους του κώδικα. Σε συστηματική μορφή, ο γεννήτορας πίνακας είναι της μορφής :

$$G = [Q^T \ \ I_{2m-m-1}\ ]\ ,$$

Όπου Q^T είναι ο ανάστροφος του Q και ο I_{2m-m-1} είναι ένας $2^m - m - 1 \ \times 2^m - m - 1$ μοναδιαίος πίνακας.

Από τη στιγμή που οι στήλες του H είναι μη μηδενικές και διακριτές, δεν υπάρχουν δύο στήλες που αν προστεθούν θα δώσουν τη μηδενική στήλη. Προκύπτει από τα προαναφερθέντα θεωρήματα ότι η ελάχιστη απόσταση σε ένα ένας Hamming κώδικα είναι τουλάχιστον 3. Αφού ο H περιέχει όλα τα μη μηδενικά m – tuples ως στήλες του, το διανυσματικό άθροισμα οποιονδήποτε δύο στηλών , έστω h_i και h_j , πρέπει να είναι επίσης μία στήλη του H έστω h_l . Επομένως :

$h_i + h_j + h_l = 0$.

Άρα η ελάχιστη απόσταση σε ένα ένας Hamming κώδικα είναι ακριβώς 3 και ο κώδικας αυτός μπορεί να διορθώνει όλα τα διανύσματα λάθους με ένα λάθος ή να ανιχνεύσει όλα τα διανύσματα λάθους με δύο ή λιγότερα λάθη.

Αν διαμορφώσουμε τον στάνταρ πίνακα για τον Hamming κώδικα μήκους $2^m - 1$,όλα τα $(2^m - 1)$ – tuples βάρους 1 μπορούν να χρησιμοποιηθούν ως coset leaders (τα διανύσματα αυτά είναι $2^m - 1$). Αφού n – k = m, ο κώδικας έχει 2^m cosets. Άρα το μηδενικό διάνυσμα **0** και τα $(2^m - 1)$ –tuples βάρους 1 είναι τα coset leaders σε ένα στάνταρ πίνακα. Ένας κώδικας διόρθωσης t – λαθών ονομάζεται τέλειος κώδικας αν ο στάνταρ πίνακας έχει όλα τα διανύσματα λάθους των t ή λιγότερων λαθών ως coset leaders και κανένα άλλο. Επομένως οι Hamming κώδικες διαμορφώνουν μία κλάση τέλειων κωδίκων που μπορούν να διορθώσουν ένα λάθος. Οι τέλειοι κώδικες είναι σπάνιοι.

Μπορούμε να διαγράψουμε οποιεσδήποτε l στήλες από τον πίνακα ελέγχου ισοτιμίας H ενός Hamming κώδικα. Αυτή η διαγραφή μας δίνει ένα m $\times 2^m - l - 1$ πίνακα H΄. Χρησιμοποιώντας τον

Η΄ ως πίνακα ελέγχου ισοτιμίας παίρνουμε ένα περιορισμένο Hamming κώδικα με τις κάτωθι παραμέτρους :

Μήκος κώδικα : $n = 2^m - 1 - 1$

Αριθμός των συμβόλων πληροφορίας : $k = 2^m - m - 1 - 1$

Αριθμός των συμβόλων ελέγχου ισοτιμίας : $n - k = m$

Ελάχιστη απόσταση : $d_{min} \geq 3$

Αν διαγράψουμε κατάλληλα στήλες από τον Η μπορούμε να πάρουμε ένα πιο σύντομο κώδικα Η με ελάχιστη απόσταση 4. Για παράδειγμά αν διαγράψουμε από τον υποπίνακα Q όλες τις στήλες με ζυγό βάρος τότε παίρνουμε ένα $m \times 2^{m-1}$ πίνακα

$$H' = [I_m \; Q']$$

Όπου ο Q΄ περιέχει $2^{m-1} - m$ στήλες μονού βάρους. Από τη στιγμή που όλες οι στήλες του Η΄ έχουν μονό βάρος δεν υπάρχουν τρεις στήλες που αν προστεθούν θα μας δώσουν το μηδενικό διάνυσμα. Όμως αν υπάρχει μια στήλη h_i με βάρος 3 στο Q΄ , θα υπάρχουν τρεις στήλες h_j , h_l και h_s στο I_m τέτοιες ώστε $h_i + h_j + h_l + h_s = 0$. Άρα για τον σύντομο κώδικα Hamming με πίνακα ελέγχου ισοτιμίας Η΄ έχει ελάχιστη απόσταση ακριβώς 4 και ο κώδικας έχει τη δυνατότητα να διορθώνει όλα τα λάθος διανύσματα με ένα λάθος και ταυτόχρονα να ανιχνεύει όλα τα λάθος διανύσματα με δύο λάθη. Όταν στην διάρκεια της μετάδοσης ενός κωδικού διανύσματος συμβεί ένα λάθος το σύνδρομο που θα προκύψει θα είναι ένα μη μηδενικό διάνυσμα και θα περιέχει

μονό αριθμό 1. Αν όμως συμβούν δύο λάθη το σύνδρομο είναι επίσης μη μηδενικό αλλά θα περιέχει ζυγό αριθμό από 1.Βασισμένοι σε αυτά τα γεγονότα η αποκωδικοποίηση γίνεται με τον ακόλουθο τρόπο :

1. Αν το σύνδρομο s είναι μηδέν υποθέτουμε ότι δεν έγινε κανένα λάθος.

2. Αν το s δεν είναι μηδέν και περιέχει μονό αριθμό από 1 τότε υποθέτουμε ότι ένα μοναδικό λάθος συνέβη. Το διάνυσμα λάθους με το ένα λάθος που αντιστοιχεί στο s προστίθεται στο λαμβανόμενο διάνυσμα για τη διόρθωση του λάθους.

3. Αν το s είναι μη μηδενικό και περιέχει ζυγό αριθμό από 1 τότε ανιχνεύεται ένα μη διορθώσιμο διάνυσμα λάθους.

Η κατανομή βάρους ενός κώδικα Hamming μήκους $n = 2^m - 1$ είναι γνωστή και ο αριθμός των κωδικών διανυσμάτων με βάρος i, A_i , είναι απλά οι συντελεστές του z^i στο ακόλουθο πολυώνυμο:

$$A(z) = \frac{1}{n+1} \{ (1+z)^n + n(1 - z)(1 - z^2)^{(n-1)/2} \}$$

Αυτό το πολυώνυμο είναι ο συντελεστής (enumerator) βάρους για τον κώδικα Hamming.

Αν ο κώδικας Hamming χρησιμοποιηθεί για έλεγχο λαθών σε ένα BSC η πιθανότητα ενός μη ανιχνεύσιμου λάθους Pu(E) που υπολογίζεται με χρήση προηγούμενων τύπων θα είναι :

$$Pu(E) = 2^{-m} \{ 1+ (2^m - 1)(1 - 2p)^{2^{m-1}} \} - (1 - p)^{2^{m-1}}$$

ΚΕΦΑΛΑΙΟ 4 - ΚΥΚΛΙΚΟΙ ΚΩΔΙΚΕΣ

Οι κυκλικοί κώδικες διαμορφώνουν μία πολύ σημαντική κατηγορία γραμμικών κωδίκων. Αυτοί οι κώδικες είναι σημαντικοί για δύο λόγους : πρώτον γιατί η κωδικοποίηση και ο υπολογισμός του συνδρόμου μπορεί να υλοποιηθεί εύκολα με καταχωρητές ολίσθησης και με συνδέσεις ανάδρασης (feedback), και δεύτερον γιατί έχουν τέτοια αλγεβρική δομή με την οποία μπορεί εύκολα να βρεθούν πολλές πρακτικές μέθοδοι για την αποκωδικοποίηση τους.

4.1 Περιγραφή των κυκλικών κωδίκων

Αν τα στοιχεία ενός n –tuple $v =(v_0,v_1,\ldots v_{n-1})$ μετακινηθούν κυκλικά μια θέση δεξιά τότε παίρνουμε ένα άλλο n –tuple :

$v^{(1)} =(v_{n-1},v_0,v_1,\ldots v_{n-2})$

που ονομάζεται κυκλική μεταφορά του **v** .Αν τα στοιχεία του **v** μετακινηθούν i φορές προς τα δεξιά τότε το αποτέλεσμα θα είναι ένα n –tuple της μορφής :

$v^{(i)} =(v_{n-i}, v_{n-i+1},\ldots, v_{n-1},v_0,v_1,\ldots v_{n-i-1})$

Είναι φανερό ότι μετακινώντας κυκλικά το **v** i θέσεις δεξιά το αποτέλεσμα είναι ισοδύναμο με το να μετακινήσουμε το **v** n – i θέσεις προς τα αριστερά.

Έτσι καταλήγουμε στον εξής ορισμό :

Ένας (n,κ) γραμμικός κώδικας C ονομάζεται **κυκλικός κώδικας** αν κάθε κυκλική μετακίνηση ενός διανύσματος του κώδικα C είναι ένα διάνυσμα που ανήκει στο κώδικα C.

Οι κυκλικοί κώδικες είναι ένα πολύ σημαντικό υποσύνολο γραμμικών κωδίκων οι οποίοι διαθέτουν πολλές χρήσιμες αλγεβρικές ιδιότητες που απλοποιούν σε μεγάλο βαθμό τις διαδικασίες κωδικοποίησης και αποκωδικοποίησης.

Πίνακας 4.1 ένας (7,4) κυκλικός κώδικας που παράγεται από το $g(x) = 1 + X + X^3$

Μήνυμα	Κωδικό διάνυσμα	Κωδικό πολυώνυμο

(0 0 0 0)	0 0 0 0 0 0 0	$0=0\cdot g(X)$
(1 0 0 0)	1 1 0 1 0 0 0	$1+X+X^3= 1\cdot g(X)$
(0 1 0 0)	0 1 1 0 1 0 0	$X+X^2+X^4= X\cdot g(X)$
(1 1 0 0)	1 0 1 1 1 0 0	$1+X^2+X^3+X^4= (1+X)\cdot g(X)$
(0 0 1 0)	0 0 1 1 0 1 0	$X^2+X^3+X^5= X^2\cdot g(X)$
(1 0 1 0)	1 1 1 0 0 1 0	$1+X+X^2+X^5= (1+X^2)\cdot g(X)$
(0 1 1 0)	0 1 0 1 1 1 0	$X+X^3+X^4+X^5= (X+X^2)\cdot g(X)$
(1 1 1 0)	1 0 0 0 1 1 0	$1+X^4+X^5= (1+X+X^2)\cdot g(X)$
(0 0 0 1)	0 0 0 1 1 0 1	$X^3+X^4+X^6= X^3\cdot g(X)$
(1 0 0 1)	1 1 0 0 1 0 1	$1+X+X^4+X^6= (1+X^3)\cdot g(X)$
(0 1 0 1)	0 1 1 1 0 0 1	$X+X^2+X^3+X^6= (X+X^3)\cdot g(X)$
(1 1 0 1)	1 0 1 0 0 0 1	$1+X^2+X^6= (1+X+X^3)\cdot g(X)$
(0 0 1 1)	0 0 1 0 1 1 1	$X^2+X^4+X^5+X^6= (X^2+X^3)\cdot g(X)$
(1 0 1 1)	1 1 1 1 1 1 1	$1+X+X^2+X^3+X^4+X^5+X^6= (1+X^2+X^3)\cdot g(X)$
(0 1 1 1)	0 1 0 0 0 1 1	$X+X^5+X^6= (X+X^2+X^3)\cdot g(X)$
(1 1 1 1)	1 0 0 1 0 1 1	$1+X^3+X^5+X^6= (1+X+X^2+X^3)\cdot g(X)$

Για την αξιοποίηση των αλγεβρικών ιδιοτήτων ενός κυκλικού κώδικα θεωρούμε τα στοιχεία του κυκλικού κώδικα $v =(v_0,v_1,\ldots v_{n-1})$ ως συντελεστές ενός πολυωνύμου ως εξής :

$$v(X) =v_0 + v_1X + v_2X^2+ \ldots+ v_{n-1} X^{n-1}$$

Με αυτό τον τρόπο κάθε διάνυσμα του κώδικα αντιστοιχεί σε ένα πολυώνυμο βαθμού n –1 ή μικρότερου. Αν το $v_{n-1} \neq 0$ ο βαθμός του v(X) είναι n –1 διαφορετικά ο βαθμός του v(X) είναι μικρότερος από n – 1. Η αντιστοιχία μεταξύ του διανύσματος v και του πολυωνύμου v(X) είναι μία προς μία. Ονομάζουμε το v(X) κωδικό πολυώνυμο του v και από το σημείο αυτό θα χρησιμοποιούμε εναλλάξ τους όρους « κωδικό διάνυσμα» και «κωδικό πολυώνυμο». Το κωδικό πολυώνυμο που αντιστοιχεί στο κωδικό διάνυσμα $v^{(i)}$ είναι :

$$v^{(i)}(X) = v_{n-i} + v_{n-i+1} X \ +\ldots+ v_{n-1}X^{i-1} + v_0X^i+v_1 X^{i+1} +\ldots+v_{n-i-1} X^{n-1}$$

Τα $v(X)$, $v^{(i)}(X)$ σχετίζονται μεταξύ τους δηλαδή πολλαπλασιάζοντας το v(X) με X^i έχουμε :

$X^i v(X) = v_0 X^i + v_1 X^{i+1} + v_2 X^{i+2} + \ldots + v_{n-1} X^{n+i-1}$

Από την παραπάνω σχέση παίρνουμε

$X^i v(X) = q(X)(X^n + 1) + v^{(i)}(X)$, (4.1)

Όπου το $q(X) = v_{n-i} + v_{n-i+1} X + \ldots + v_{n-1} X^{i-1}$

Συμπεραίνουμε λοιπόν ότι το κωδικό πολυώνυμο $v^{(i)}(X)$ είναι απλώς το υπόλοιπο της διαίρεσης του πολυωνύμου $X^i v(X)$ με το $X^n + 1$.

Στο σημείο αυτό παραθέτονται μερικά χρήσιμα θεωρήματα που σχετίζονται με τις αλγεβρικές ιδιότητες ενός κυκλικού κώδικα με τις οποίες απλοποιούνται σε μεγάλο βαθμό οι διαδικασίες κωδικοποίησης και υπολογισμού του συνδρόμου.

Θεώρημα 1 Το μη μηδενικό κωδικό πολυώνυμο ελάχιστου βαθμού σε ένα κυκλικό κώδικα C είναι μοναδικό.

Θεώρημα 2 Έστω $g(X) = g_0 + g_1 X + \ldots + g_{r-1} X^{r-1} + X^r$ το μη μηδενικό πολυώνυμο ελάχιστου βαθμού σε ένα (n,k) κυκλικού κώδικα C. Τότε ο σταθερός όρος g_0 πρέπει να ισούται με 1.

Από το θεώρημα αυτό προκύπτει ότι το μη μηδενικό πολυώνυμο ελάχιστου βαθμού σε ένα (n,k) κυκλικού κώδικα C θα έχει τη μορφή $g(X) = 1 + g_1 X + g_2 X^2 + \ldots + g_{r-1} X^{r-1} + X^r$

Θεωρούμε τα πολυώνυμα $Xg(X)$, $X^2 g(X)$, ..., $X^{n-r-1} g(X)$ που έχουν βαθμό r+1, r+2,…, n – 1 αντίστοιχα. Από τη σχέση (4.1) προκύπτει ότι :

$Xg(X) = g^{(1)}(X)$, $X^2 g(X) = g^{(2)}(X)$,..., $X^{n-r-1} g(X) = g^{(n-r-1)}(X)$ είναι δηλαδή κυκλικές μετακινήσεις του κωδικού πολυωνύμου g(X) επομένως είναι όλα κωδικά πολυώνυμα του C. Από τη στιγμή που ο κώδικας C είναι γραμμικός, ένας γραμμικός συνδυασμός των $g(X)$,$Xg(X)$, $X^2 g(X)$,..., $X^{n-r-1} g(X)$ το πολυώνυμο :

$v(X) = v_0 g(X) + v_1 Xg(X) + \ldots + v_{n-r-1} X^{n-r-1} g(X) = (v_0 + v_1 X + \ldots + v_{n-r-1} X^{n-r-1}) g(X)$

θα είναι επίσης ένα κωδικό πολυώνυμο όπου $v_i = 0$ ή 1.

Θεώρημα 3 έστω $g(X) = 1+ g_1X+ g_2X^2+\ldots+ g_{r-1}X^{r-1} + X^r$ ένα μη μηδενικό κωδικό πολυώνυμο ελάχιστου βαθμού σε ένα (n,k) κυκλικό κώδικα C. Ένα δυαδικό πολυώνυμο βαθμού n – 1 ή μικρότερου είναι κωδικό πολυώνυμο αν και μόνο αν είναι πολλαπλάσιο του g(X).

Ο αριθμός των δυαδικών πολυωνύμων βαθμού n – 1 ή μικρότερου που είναι πολλαπλάσια του g(X) είναι 2^{n-r} . Αυτά τα πολυώνυμα διαμορφώνουν όλα τα κωδικά πολυώνυμα του (n,k) κυκλικού κώδικα C. Από τη στιγμή που υπάρχουν $2^κ$ κωδικά πολυώνυμα στο C τα 2^{n-r} πρέπει να ισούται με $2^κ$.Επομένως το r = n – k και από αυτό προκύπτει ότι ο βαθμός του g(X) είναι n – k και το g(X) θα έχει τη μορφή : $g(X) = 1+ g_1X+ g_2X^2+\ldots+ g_{n-k-1}X^{n-k-1} + X^{n-k}$

Από τα παραπάνω προκύπτει το εξής θεώρημα :

Θεώρημα 4 Σε ένα (n,k) κυκλικό κώδικα C υπάρχει ένα μοναδικό κωδικό πολυώνυμο βαθμού

n – k το $g(X) = 1+ g_1X+ g_2X^2+\ldots+ g_{n-k-1}X^{n-k-1} + X^{n-k}$. Κάθε κωδικό πολυώνυμο είναι πολλαπλάσιο του g(X) και κάθε δυαδικό πολυώνυμο βαθμού n – 1 ή μικρότερου που είναι πολλαπλάσιο του g(X) είναι κωδικό πολυώνυμο.

Έτσι λοιπόν κάθε κωδικό πολυώνυμο v(X)του κυκλικού κώδικα μπορεί να εκφραστεί ως

$v(X) = u(X)g(X) = (u_0 + u_1 X+ \ldots+ u_{k-1} X^{k-1}) g(X)$ όπου κ είναι ο αριθμός των ψηφίων πληροφορίας που πρόκειται να κωδικοποιηθούν. Το g(X) ονομάζεται γενεσιουργό (generator) πολυώνυμο του κώδικα.

Θεώρημα 5 Το γενεσιουργό (generator) πολυώνυμο g(X) ενός (n,k) κυκλικού κώδικα C είναι παράγοντας του

X^n+1 $[X^n+1 = \{ X^κ+α(X)\} g(X)]$.

Το ερώτημα που προκύπτει είναι αν για κάθε n,k υπάρχει ένας (n,k) κυκλικός κώδικας C. Σαν απάντηση δίνεται το παρακάτω θεώρημα.

Θεώρημα 6 Αν g(X) είναι ένα πολυώνυμο βαθμού n – k και είναι παράγοντας του X^n+1 τότε το g(X) παράγει ένα (n,k) κυκλικό κώδικα.

Όταν δίνεται το γενεσιουργό (generator) πολυώνυμο σε ένα (n,k) κυκλικό κώδικα C, ο κώδικάς μπορεί να αποκτήσει συστηματική μορφή (δηλαδή τα κ ψηφία πληροφορίας να βρίσκονται όλα στα δεξιά και τα n – k ψηφία ελέγχου ισοτιμίας όλα αριστερά) ακολουθώντας τα εξής βήματα :

Βήμα 1 Πολλαπλασιάζουμε αρχικά το u (X)(το πολυώνυμο που αντιστοιχεί στο διάνυσμα προς κωδικοποίηση) με X^{n-k}

Βήμα 2 βρίσκουμε το υπόλοιπο b(X) (τα ψηφία ελέγχου ισοτιμίας) από τη διαίρεση του

X^{n-k} u (X) με το γενεσιουργό (generator) πολυώνυμο g(X).

Βήμα 3 Προσθέτουμε το b(X) με το X^{n-k} u (X) και παίρνουμε έτσι το κωδικό πολυώνυμο

b(X) +X^{n-k} u (X) .

4.2 Γεννήτορες πίνακες και πίνακες ελέγχου ισοτιμίας για κυκλικούς κώδικες

Θεωρούμε ένα (n,k) κυκλικό κώδικα C με γενεσιουργό (generator) πολυώνυμο $g(X)= g_0+ g_1X+ \ldots+ g_{n-k} X^{n-k}$. Στην προηγούμενη παράγραφο δείξαμε ότι τα πολυώνυμα g(X),Xg(X), X^2 g(X), ..., $X^{\kappa-1}$ g(X) διαμορφώνουν τον κώδικα C.Αν τα κ n –tuples που αντιστοιχούν σε αυτά τα κ κωδικά πολυώνυμα χρησιμοποιηθούν σαν γραμμές σε ένα κ ×κ πίνακα τότε παίρνουμε το γεννήτορα πίνακα του κώδικα C :

$$\begin{bmatrix} g_0 & g_1 & g_2 & . & . & . & . & . & g_{n-k} & 0 & 0 & 0 & . & . & 0 \\ 0 & g_0 & g_1 & g_2 & . & . & . & . & . & g_{n-k} & 0 & 0 & . & . & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & . & . & . & . & . & g_{n-k} & 0 & . & . & 0 \\ . & & & & & & & & & & & & & & . \\ . & & & & & & & & & & & & & & . \\ . & & & & & & & & & & & & & & . \\ 0 & 0 & . & . & . & 0 & g_0 & g_1 & g_2 & . & . & . & . & . & g_{n-k} \end{bmatrix}$$

(Προσοχή το $g_0 = g_{n-\kappa} = 1$).Γενικώς το G δεν είναι σε συστηματική μορφή αλλά με διάφορες πράξεις μεταξύ των γραμμών μπορούμε να εμφανίσουμε τον πίνακα σε συστηματική μορφή. Ο νέος πίνακας σε συστηματική μορφή θα είναι επίσης γεννήτορας πίνακας του ίδιου κώδικα όπως ο G. Γνωρίζουμε ότι το γενεσιουργό (generator) πολυώνυμο είναι παράγοντας του $X^n + 1$ δηλαδή :

$X^n + 1 = g(X)\mathbf{h}(X)$ όπου το πολυώνυμο $\mathbf{h}(X)$ είναι βαθμού κ και έχει την ακόλουθη μορφή :

$\mathbf{h}(X) = \mathbf{h}_0 + \mathbf{h}_1 X + \ldots + \mathbf{h}_k X^k$ με $\mathbf{h}_0 = \mathbf{h}_k = 1$.

Το πολυώνυμο $\mathbf{h}(X)$ χρησιμοποιείται για τη δημιουργία του πίνακα ελέγχου ισοτιμίας του κώδικα C. Έστω $v = (v_0, v_1, \ldots v_{n-1})$ είναι ένα κωδικό διάνυσμα του C. Τότε το $v(X) = a(X)g(X)$. Πολλαπλασιάζοντας το $v(X)$ με το $\mathbf{h}(X)$ έχουμε :

$v(X)\,\mathbf{h}(X) = a(X)g(X)\,\mathbf{h}(X) = a(X)(X^n + 1) = a(X) + X^n a(X)$

Αφού ο βαθμός του $a(X)$ είναι $\kappa - 1$ ή λιγότερο οι δυνάμεις X^k, $X^{k+1}, \ldots, X^{n-1}$ δεν πρέπει να εμφανίζονται στο $a(X) + X^n a(X)$. Επομένως στο γινόμενο $v(X)\,\mathbf{h}(X)$ οι συντελεστές των X^k, $X^{k+1}, \ldots, X^{n-1}$ πρέπει να είναι μηδέν και έτσι παίρνουμε τις εξής $n - k$ ισότητες :

$$\sum_{i=0}^{k} h_i\, u_{n-i-j} = 0 \quad \text{για } 1 \le j \le n-k \qquad (4.2)$$

Ο αντίστροφος του $\mathbf{h}(X)$:

$X^\kappa \mathbf{h}(X^{-1}) = \mathbf{h}_\kappa + \mathbf{h}_{\kappa-1} X + \mathbf{h}_{\kappa-2} X^2 + \ldots + \mathbf{h}_0 X^\kappa$

Το $X^\kappa \mathbf{h}(X^{-1})$ είναι παράγοντας του $X^n + 1$. Επομένως το πολυώνυμο $X^\kappa \mathbf{h}(X^{-1})$ παράγει ένα $(n, n-k)$ κυκλικό κώδικα με τον ακόλουθο $(n-k) \times n$ πίνακα ως γεννήτορα πίνακα.

$$H=\begin{bmatrix} h_k & h_{k-1} & h_{k-2} & . & . & . & . & . & h_0 & 0 & . & . & . & . & 0 \\ 0 & h_k & h_{k-1} & h_{k-2} & . & . & . & . & . & h_0 & 0 & . & . & . & 0 \\ 0 & 0 & h_k & h_{k-1} & h_{k-2} & . & . & . & . & . & h_0 & . & . & . & 0 \\ . & & & & & & & & & & & & & & . \\ . & & & & & & & & & & & & & & . \\ . & & & & & & & & & & & & & & . \\ 0 & 0 & . & . & . & 0 & h_k & h_{k-1} & h_{k-2} & . & . & . & . & . & h_0 \end{bmatrix}$$

Από τις n –k εξισώσεις που περιγράφονται από τη σχέση (4.2)προκύπτει ότι κάθε κωδικό διάνυσμα v του C είναι ορθογώνιο με κάθε γραμμή του **H**. Επομένως ο **H** είναι ο πίνακας ελέγχου ισοτιμίας του κώδικα C και ο χώρος των γραμμών του **H** είναι ο δυαδικός κώδικας του C. Το πολυώνυμο **h**(X) ονομάζεται πολυώνυμο ισοτιμίας του χώρου C (ο χώρος C ορίζεται μοναδικά από το πολυώνυμο ισοτιμίας).

Θεώρημα 7 Έστω C είναι ένας (n,k) κυκλικός κώδικας με γενεσιουργό (generator) πολυώνυμο g(X).Ο δυαδικός κώδικας του C είναι επίσης ένας κυκλικός κώδικας και παράγεται από το πολυώνυμο $X^{\kappa}\,\mathbf{h}(X^{-1})$ όπου $\mathbf{h}(X) = (X^{n}+1) / g(X)$.

Για να γραφτεί ο γεννήτορας πίνακας σε συστηματική μορφή διαιρούμε το $X^{n-\kappa-i}$με το γενεσιουργό (generator) πολυώνυμο για i= 0,1,…,k – 1 και παίρνουμε

$X^{n-\kappa-i} = a_i(X)g(X) + b_i(X)$ όπου

$b_i(X) = b_{i0} + b_{i1}X +\ldots+ b_{i,n-k-1}\,X^{n-\kappa-1}$

Από τη στιγμή που το $b_i(X) + X^{n-\kappa-i}$ για i= 0,1,…,k – 1είναι πολλαπλάσια του g(X) είναι κωδικά πολυώνυμα. Χρησιμοποιώντας αυτά τα κ πολυώνυμα ως γραμμές σε ένα κ × n πίνακα παίρνουμε τον γεννήτορα πίνακα G σε συστηματική μορφή.

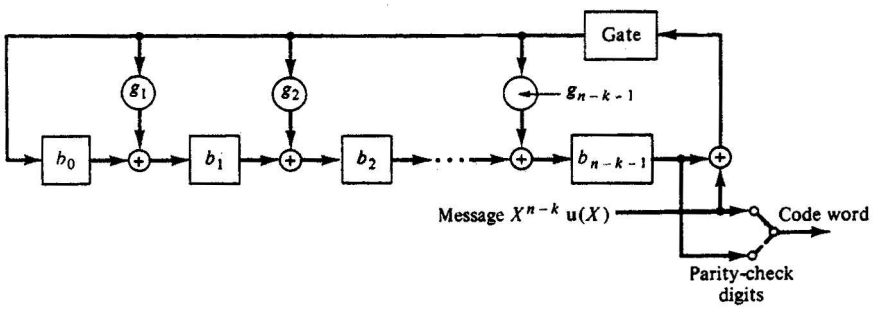
4.3 Κωδικοποίηση των κυκλικών κωδίκων

Η κωδικοποίηση ενός (n, k) κυκλικού κώδικα αποτελείται από 3 βήματα (1) πολλαπλασιασμός του πολυωνύμου του μηνύματος u(X) με X^{n-k} (2)διαίρεση του X^{n-k} u(X) με το g(X) για την εύρεση του υπόλοιπου b(X) και (3) διαμόρφωση της κωδικής λέξης $b(X) + X^{n-k}$ u(X). Τα τρία αυτά βήματα

μπορούν εύκολα να υλοποιηθούν από ένα κύκλωμα διαίρεσης που είναι ένας γραμμικός (n- k) καταχωρητής ολίσθησης με συνδέσεις ανάδρασης βασισμένες στο γενεσιουργό (generator) πολυώνυμο

$g(X) = 1+ g_1X+ g_2X^2+\ldots+ g_{n-k-1}X^{n-k-1} + X^{n-k}$.

Ένα τέτοιο κύκλωμα φαίνεται στο σχήμα 4.1. Η διαδικασία κωδικοποίησης υλοποιείται ως εξής :



Σχήμα 4.1 Κύκλωμα κωδικοποίησης για ένα (n,k) κυκλικό κώδικα με γεννεσιουργό πολυώνυμο

$g(X) = 1+ g_1X+ g_2X^2+\ldots+ g_{n-k-1}X^{n-k-1} + X^{n-k}$.

Βήμα 1 Με την πύλη ανοικτή τα κ ψηφία πληροφορίας $u_0, u_1,\ldots, u_{k-1}$,ολισθαίνουν στο κύκλωμα και ταυτόχρονα στο κανάλι επικοινωνίας. Ολισθαίνοντας το μήνυμα στο κύκλωμα από το μπροστινό άκρο ισοδυναμεί με τον προπολλαπλασιασμό του u(X) με το X^{n-k} . Μόλις ολόκληρο το μήνυμα εισέλθει στο κύκλωμα τα n –k ψηφία που βρίσκονται μέσα στο καταχωρητή διαμορφώνουν το υπόλοιπο και επομένως τα ψηφία ελέγχου ισοτιμίας.

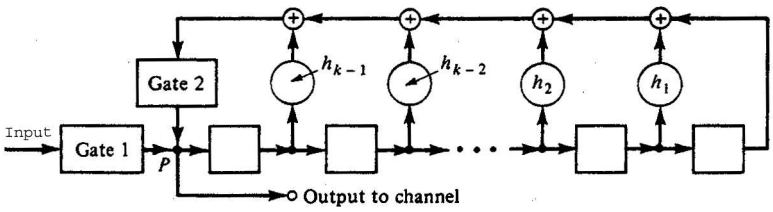
Βήμα 2 Διακόπτουμε τη σύνδεση ανάδρασης με το να σβήσουμε την πύλη.

Βήμα 3 Ολισθαίνουν τα ψηφία ελέγχου ισοτιμίας και τα στέλνουμε στο κανάλι. Αυτά τα n –k ψηφία ελέγχου ισοτιμίας $b_0,b_1,..,b_{n-k-1}$ μαζί με τα κ ψηφία πληροφορίας αποτελούν ένα ολόκληρο κωδικό διάνυσμα.

Η κωδικοποίηση ενός κυκλικού κώδικα μπορεί να επιτευχθεί και με τη χρήση του πολυωνύμου ισοτιμίας $\mathbf{h}(X) = \mathbf{h}_0+ \mathbf{h}_1X+\ldots+ \mathbf{h}_k X^k$ με $\mathbf{h}_0 = \mathbf{h}_k =1$. Έστω $v = (v_0, v_1,\ldots, v_{n-1})$ είναι ένα κωδικό διάνυσμα. Τα στοιχεία του v τότε θα ικανοποιούν τις παρακάτω εξισώσεις :

$$v_{n-k-j} = \sum_{i=0}^{k-1} h_i \, v_{n-i-j} \quad \text{για } 1 \le j \le n-k$$

που είναι γνωστές ως διαφορικές εξισώσεις. Για ένα κυκλικό κώδικα σε συστηματική μορφή τα στοιχεία $v_{n-κ}$, $v_{n-κ+1}$,..., v_{n-1} για κάθε κωδικό διάνυσμα είναι τα ψηφία πληροφορίας. Όταν δίνονται τα κ ψηφία πληροφορίας το επόμενο βήμα είναι ο καθορισμός των n-k ψηφία ισοτιμίας ελέγχου v_0 , v_1,..., $v_{n-κ-1}$. Το κύκλωμα κωδικοποίησης που βασίζεται σε αυτή τη μέθοδο δίνεται στο σχήμα 4.2.



Σχήμα 4.2 Κύκλωμα κωδικοποίησης για ένα (n,k) κυκλικό κώδικα βασισμένο πολυώνυμο ισοτιμίας

$$\mathbf{h}(X) = \mathbf{1} + \mathbf{h_1}X + \ldots + X^k$$

Οι συνδέσεις ανάδρασης βασίζονται στα στοιχεία του πολυωνύμου ελέγχου ισοτιμίας **h**(X). Η διαδικασία κωδικοποίησης αποτελείται από τέσσερα βήματα

Βήμα 1 Η πύλη 1 είναι αρχικά ανοικτή και η πύλη 2 είναι κλειστή. Τα κ ψηφία πληροφορίας που καθορίζονται από το $\mathbf{h}(X) = \mathbf{h}_0 + \mathbf{h_1}X + \ldots + \mathbf{h}_k X^k$ ολισθαίνουν στον καταχωρητή και ταυτόχρονα στο κανάλι επικοινωνίας.

Βήμα 2 Μόλις τα κ ψηφία πληροφορίας μπουν στον καταχωρητή ολίσθησης η πύλη 1 σβήνει και ανοίγει η πύλη 2.Το 1ο ψηφίο ελέγχου ισοτιμίας θα δίνεται

$$v_{n-κ-1} = \mathbf{h}_0 v_{n-1} + \mathbf{h_1} v_{n-2} + \ldots + \mathbf{h}_k v_{n-k} = u_{k-1} + \mathbf{h_1} u_{k-2} + \ldots + \mathbf{h}_{k-1} u_0$$

Βήμα 3 Ο καταχωρητής ολισθαίνει μία φορά. Το πρώτο ψηφίο ελέγχου ισοτιμίας μεταφέρεται στο κανάλι και στον καταχωρητή. Το 2ο ψηφίο ελέγχου ισοτιμίας θα δίνεται

$$v_{n-κ-2} = \mathbf{h}_0 v_{n-2} + \mathbf{h_1} v_{n-3} + \ldots + \mathbf{h}_{k-1} v_{n-k-1} = u_{k-2} + \mathbf{h_1} u_{k-3} + \ldots + \mathbf{h}_{k-2} u_0 + \mathbf{h}_{k-1} u_{v-κ-1}$$

Βήμα 4 Το βήμα 3 επαναλαμβάνεται μέχρι ότου διαμορφωθούν τα n-k ψηφία ελέγχου ισοτιμίας και μεταφερθούν στο κανάλι.

Το παραπάνω κύκλωμα κωδικοποίησης είναι γνωστό ως ένας κ – επίπεδος καταχωρητής .Το κύκλωμα αυτό συνίσταται για κώδικες με περισσότερα ψηφία ελέγχου ισοτιμίας από τα ψηφία μηνύματος γιατί είναι πιο οικονομικό διαφορετικά προτιμάται το (n-k) – επίπεδο κύκλωμα κωδικοποίησης.

4.4 Υπολογισμός του συνδρόμου και ανίχνευση λαθών

Έστω r το λαμβανόμενο διάνυσμα. Εξαιτίας της παρουσίας θορύβου στο κανάλι το λαμβανόμενο διάνυσμα μπορεί να διαφέρει από μεταδιδόμενο κωδικό διάνυσμα. Αν ο κώδικας είναι γραμμικός το πρώτο βήμα αποκωδικοποίησης είναι ο υπολογισμός του συνδρόμου $s = r \cdot H^T$ όπου **H** είναι ο πίνακας ελέγχου ισοτιμίας. Αν το σύνδρομο είναι μηδέν τότε το r είναι κωδικό διάνυσμα και ο αποκωδικοποιητής δέχεται το r ως το μεταδιδόμενο κωδικό διάνυσμα. Αν το σύνδρομο δεν είναι μηδέν τότε το r δεν είναι κωδικό διάνυσμα και η παρουσία λαθών ανιχνεύεται.

Έχει δειχτεί ότι σε ένα γραμμικό συστηματικό κώδικα το σύνδρομο είναι απλώς το διανυσματικό άθροισμα των λαμβανόμενων ψηφίων ισοτιμίας και των ψηφίων ελέγχου ισοτιμίας που υπολογίζονται από τα λαμβανόμενα ψηφία πληροφορίας. Για ένα κυκλικό κώδικα που είναι σε συστηματική μορφή το σύνδρομο μπορεί να υπολογιστεί εύκολα. Το λαμβανόμενο διάνυσμα r χρησιμοποιείται σαν ένα πολυώνυμο βαθμού n –1 ή μικρότερου,

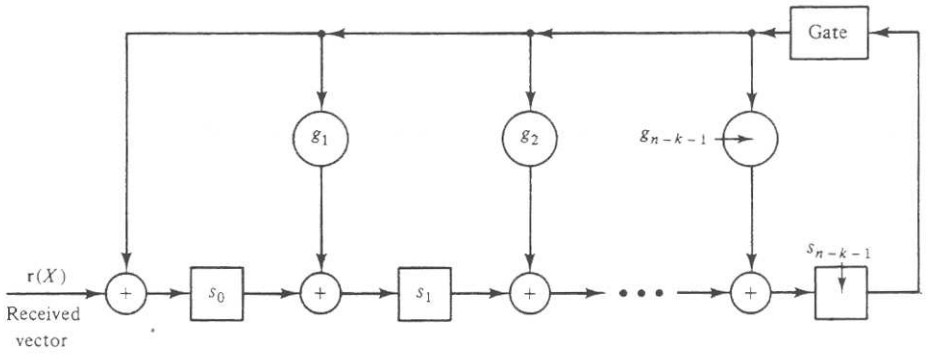
$r(X) = r_0 + r_1X + r_2 X^2 + \ldots + r_{n-1} X^{n-1}$

Διαιρώντας το r(X) με το γενεσιουργό (generator) πολυώνυμο g(X) έχουμε :

$r(X) = a(X)\, g(X) + s(X)$

Το υπόλοιπο s(X) είναι ένα πολυώνυμο βαθμού n – κ – 1 ή μικρότερου. Οι n – κ συντελεστές του s(X) διαμορφώνουν το σύνδρομο s. Είναι φανερό ότι το s(X) ισούται με το μηδέν αν και μόνο αν το λαμβανόμενο πολυώνυμο r(X) είναι ένα κωδικό πολυώνυμο. Από το σημείο αυτό θα ονομάζουμε το s(X) σύνδρομο. Ο υπολογισμός του συνδρόμου μπορεί να γίνει από ένα κύκλωμα διαίρεσης που φαίνεται στο σχήμα 4.3 και που είναι ίδιο με το (n – κ) –επίπεδο κύκλωμα

κωδικοποίησης με τη διαφορά ότι το λαμβανόμενο πολυώνυμο r(X) ολισθαίνει στον καταχωρητή από το αριστερό άκρο. Το λαμβανόμενο πολυώνυμο r(X) ολισθαίνει στον καταχωρητή αφού πρώτα έχει τεθεί στο μηδέν. Μόλις ολόκληρο το r(X) μεταφερθεί στον καταχωρητή τα περιεχόμενα του διαμορφώνουν το σύνδρομο s(X).



Σχήμα 4.3 Ένα κύκλωμα συνδρόμου με (n – k) επίπεδα με την είσοδο από το αριστερό άκρο

Λόγω της κυκλικής δομής του κώδικα το σύνδρομο s(X) έχει την ακόλουθη ιδιότητα :

Θεώρημα 8 Έστω s(X) το σύνδρομο του λαμβανόμενου πολυώνυμου $r(X) = r_0 + r_1X + r_2 X^2 + \ldots + r_{n-1} X^{n-1}$. Τότε το υπόλοιπο $s^{(1)}(X)$ που προκύπτει από τη διαίρεση του X s(X) με το γενεσιουργό (generator) πολυώνυμο g(X) είναι το σύνδρομο του $r^{(1)}(X)$ που είναι η πρώτη κυκλική ολίσθηση του r(X).

Από το θεώρημα αυτό προκύπτει ότι το υπόλοιπο $s^{(i)}(X)$ που προκύπτει από τη διαίρεση του

X^i s(X) με το γενεσιουργό (generator) πολυώνυμο g(X) είναι το σύνδρομο του $r^{(i)}(X)$ που είναι η i κυκλική ολίσθηση του r(X). Αυτή η ιδιότητα είναι χρήσιμη στην αποκωδικοποίηση των κυκλικών κωδίκων. Για να πάρουμε το σύνδρομο $s^{(i)}(X)$ απλώς μετακινούμε τον καταχωρητή του συνδρόμου i φορές με s(X) τα αρχικά περιεχόμενα.

Έστω v(X) το μεταδιδόμενο διάνυσμα και $e(x) = e_0 + e_1X + \ldots + e_{n-1} X^{n-1}$ είναι το διάνυσμα λάθους (error pattern). Τότε το λαμβανόμενο πολυώνυμο είναι

$r(X) = v(X) + e(X)$

Αφού το v(X) είναι πολλαπλάσιο του g(X) έχουμε την ακόλουθη σχέση μεταξύ του διανύσματος λάθους και του συνδρόμου :

$e(X) = [a(X) + b(X)]g(X) + s(X)$

όπου $b(X)g(X) = v(X)$. Από αυτή τη σχέση φαίνεται καθαρά ότι το σύνδρομο είναι το υπόλοιπο που προκύπτει από τη διαίρεση του διανύσματος λάθους με το γενεσιουργό (generator) πολυώνυμο. Το σύνδρομο μπορεί να υπολογιστεί από το λαμβανόμενο διάνυσμα , το διάνυσμα λάθους όμως είναι άγνωστο στον αποκωδικοποιητή. Επομένως ο αποκωδικοποιητής πρέπει να εντοπίσει το διάνυσμα λάθους βασισμένος στο σύνδρομο. Αν το e(X) είναι coset leader του στάνταρ πίνακα και αν πραγματοποιηθεί αποκωδικοποίηση table – lookup το e(X) θα αναγνωριστεί σωστά από το σύνδρομο. Το s(X) είναι μηδέν αν και μόνο αν ή το διάνυσμα λάθους είναι μηδέν ή είναι παρόμοιο με ένα κωδικό διάνυσμα. Αν το e(X) είναι παρόμοιο με ένα κωδικό διάνυσμα τότε το e(X) είναι ένα μη ανιχνεύσιμο διάνυσμα λάθους. Οι κυκλικοί κώδικες είναι πολύ ευαίσθητοι στην ανίχνευση λαθών. Το κύκλωμα ανίχνευσης λαθών είναι απλώς ένα κύκλωμα συνδρόμου με μία OR πύλη που έχει σαν εισόδους τα ψηφία του συνδρόμου. Αν το σύνδρομο δεν είναι μηδέν τότε η έξοδος της OR πύλης είναι «1» και η παρουσία λαθών ανιχνεύεται.

Στο ερώτημα του ποια είναι η δυνατότητα ανίχνευσης λαθών η απάντηση δίνεται με το παρακάτω θεώρημα:

Θεώρημα 9 Ένας (n,k) κυκλικός κώδικας είναι ικανός στο να ανιχνεύει κάθε λάθος «ξέσπασμα » (error burst δηλαδή λάθη που είναι περιορισμένα σε συνεχείς θέσεις) μήκους n – k ή λιγότερα συμπεριλαμβανομένου και των λαθών που βρίσκονται σε συνεχόμενες θέσεις στα άκρα του μηνύματος.(Στην πραγματικότητα ένα μεγάλο ποσοστό «ξεσπάσματος » λαθών μήκους n – k +1 ή μεγαλύτερα μπορεί να ανιχνευθούν).

Το όριο των μη ανιχνεύσιμων λαθών δίνεται από το παρακάτω θεώρημα :

Θεώρημα 10 Το κλάσμα (όριο)των μη ανιχνεύσιμων λαθών («ξέσπασμα ») μήκους n – k +1 είναι

$2^{-(n-k-1)}$.

Για $l > n-k+1$ υπάρχουν 2^{l-2} «ξεσπάσματα » μήκους l που ξεκινούν από τη θέση του ith ψηφίου και τελειώνουν στην θέση (i + l – 1) th θέση ψηφίου. Μεταξύ αυτών των «ξεσπασμάτων» αυτά που δεν ανιχνεύονται είναι όσα έχουν την παρακάτω μορφή :

$e(X) = X^i a(X)\, g(X)$

$a(X) = a_0 + a_1 X + \ldots + a_{l-(n-k)-1} X^{l-(n-k)-1}$ με $a_0 = a_{l-(n-k)-1} = 1$

Θεώρημα 11 Για $l > n-k+1$το κλάσμα των μη ανιχνεύσιμων λαθών («ξέσπασμα ») μήκους l είναι $2^{-(n-k)}$.

4.5 Αποκωδικοποίηση των κυκλικών κωδίκων

Η αποκωδικοποίηση των κυκλικών κωδίκων περιλαμβάνει τρία βήματα :

1) υπολογισμός συνδρόμου

2)συσχέτιση συνδρόμου με ένα διάνυσμα λάθους (error pattern)

3) διόρθωση λάθους

Στην παράγραφο 4.4 είδαμε ότι ο υπολογισμός του συνδρόμου για κυκλικούς κώδικες μπορεί να επιτευχθεί με ένα κύκλωμα διαίρεσης που η πολυπλοκότητα του είναι γραμμικώς ανάλογη με τον αριθμό των ψηφίων ελέγχου ισοτιμίας (δηλαδή με το n – k). Η διαδικασία της διόρθωσης λαθών είναι απλώς η πρόσθεση του διανύσματος λάθους με το λαμβανόμενο διάνυσμα. Αυτό μπορεί να επιτευχθεί με μία απλή αποκλειστική OR πύλη αν η διόρθωση γίνεται με σειριακό τρόπο (δηλαδή ένα ψηφίο κάθε φορά) ή με n αποκλειστικές OR πύλες αν η διόρθωση γίνεται με παράλληλο τρόπο. Η συσχέτιση του συνδρόμου με ένα διάνυσμα λάθους μπορεί να γίνει μέσω ενός πίνακα αποκωδικοποίησης. Μια άμεση προσέγγιση σχεδιασμού ενός κυκλώματος αποκωδικοποίησης είναι μέσω ενός συνδυαστικού λογικού κυκλώματος που υλοποιεί μία διαδικασία ελέγχου πίνακα (table lookup). Όμως το όριο αυτής της προσέγγισης είναι η πολυπλοκότητα αυτού του κυκλώματος που τείνει να αυξάνεται εκθετικά σε σχέση με το μήκος του κώδικα και με τον αριθμό λαθών που

σκοπεύει να διορθώσει. Οι κυκλικοί κώδικες διαθέτουν αρκετές αλγεβρικές και γεωμετρικές ιδιότητες που αν χρησιμοποιηθούν σωστά είναι πολύ πιθανή η απλοποίηση του κυκλώματος αποκωδικοποίησης.

Η κυκλική δομή ενός κυκλικού κώδικα επιτρέπει την αποκωδικοποίηση του λαμβανόμενου διανύσματος $r(X) = r_0+r_1X+r_2X^2+\ldots+r_{n-1}X^{n-1}$ με σειριακό τρόπο. Τα λαμβανόμενα ψηφία αποκωδικοποιούνται ένα κάθε φορά και όλα τα ψηφία αποκωδικοποιούνται στο ίδιο κύκλωμα. Μόλις το σύνδρομο υπολογιστεί το κύκλωμα αποκωδικοποίησης ελέγχει αν το σύνδρομο s(X) αντιστοιχεί σε ένα διορθώσιμο error pattern $e(x) = e_0+ e_1X+ \ldots+ e_{n-1} X^{n-1}$ με το λάθος να βρίσκεται στη θέση υψηλότερης τάξης X^{n-1} (e_{n-1} =1).Αν το s(X) δεν αντιστοιχεί σε ένα error pattern με e_{n-1} =1 , το λαμβανόμενο πολυώνυμο (που είναι αποθηκευμένο σε ένα καταχωρητή – buffer) και ο καταχωρητής του συνδρόμου μετακινούνται κυκλικά μια φορά ταυτόχρονα. Έτσι λαμβάνουμε το $r^{(1)}(X) = r_{n-1}+ r_0X+r_1X^2+\ldots+r_{n-2}X^{n-1}$ και τα νέα περιεχόμενα του καταχωρητή του συνδρόμου διαμορφώνουν το σύνδρομο $s^{(1)}(X)$ του $r^{(1)}(X)$.Με αυτό τον τρόπο το δεύτερο ψηφίο του r(X) το r_{n-2} γίνεται το πρώτο στοιχείο του $r^{(1)}(X)$.Το ίδιο κύκλωμα αποκωδικοποίησης θα ελέγξει αν το $s^{(1)}(X)$ αντιστοιχεί σε ένα error pattern με λάθος στη θέση X^{n-1}.

Αν το σύνδρομο s(X) του r(X) αντιστοιχεί σε ένα error pattern με λάθος στη θέση X^{n-1} το πρώτο λαμβανόμενο ψηφίο r_{n-1} είναι ένα λάθος ψηφίο και πρέπει να διορθωθεί. Η διόρθωση γίνεται παίρνοντας το άθροισμα $r_{n-1} \oplus e_{n-1}$. Η διόρθωση αυτή μας δίνει ένα τροποποιημένο πολυώνυμο που συμβολίζεται με $r_1(X) = r_0+r_1X+r_2X^2+\ldots+r_{n-2}X^{n-2} +(r_{n-1} \oplus e_{n-1}) X^{n-1}$.

Έτσι εξαλείφεται η επίδραση του λάθους ψηφίου e_{n-1} από το σύνδρομο s(X). Αυτό επιτυγχάνεται με το να προσθέσουμε το $e'(x) = X^{n-1}$ στο s(X). Αυτό το άθροισμα είναι το σύνδρομο του τροποποιημένου λαμβανόμενου πολυώνυμου $r_1(X)$.Αν τώρα μετακινήσουμε κυκλικά το $r_1(X)$ και τον καταχωρητή του συνδρόμου μια φορά ταυτόχρονα θα πάρουμε το πολυώνυμο $r_1^{(1)}(X) = (r_{n-1} \oplus e_{n-1}) ++ r_0X+\ldots+r_{n-2}X^{n-1}$. Το σύνδρομο $s_1^{(1)}(X)$ του $r_1^{(1)}(X)$ είναι το υπόλοιπο της διαίρεσης του $X[s(X) +X^{n-1}]$ με το γενεσιουργό (generator) πολυώνυμο g(X) και θα έχουμε $s_1^{(1)}(X) = s^{(1)}(X) +1$.

Επομένως αν 1 προστίθεται στο αριστερό άκρο του καταχωρητή του συνδρόμου ενώ έχει μετακινηθεί τότε παίρνουμε το $s_1^{(1)}(X)$.Το κύκλωμα αποκωδικοποίησης τότε θα προχωρήσει στην αποκωδικοποίηση του επόμενου λαμβανόμενου ψηφίου r_{n-2}με τον ίδιο τρόπο. Κάθε φορά που ένα λάθος ανιχνεύεται και διορθώνεται η επίδραση του εξαλείφεται από το σύνδρομο. Η

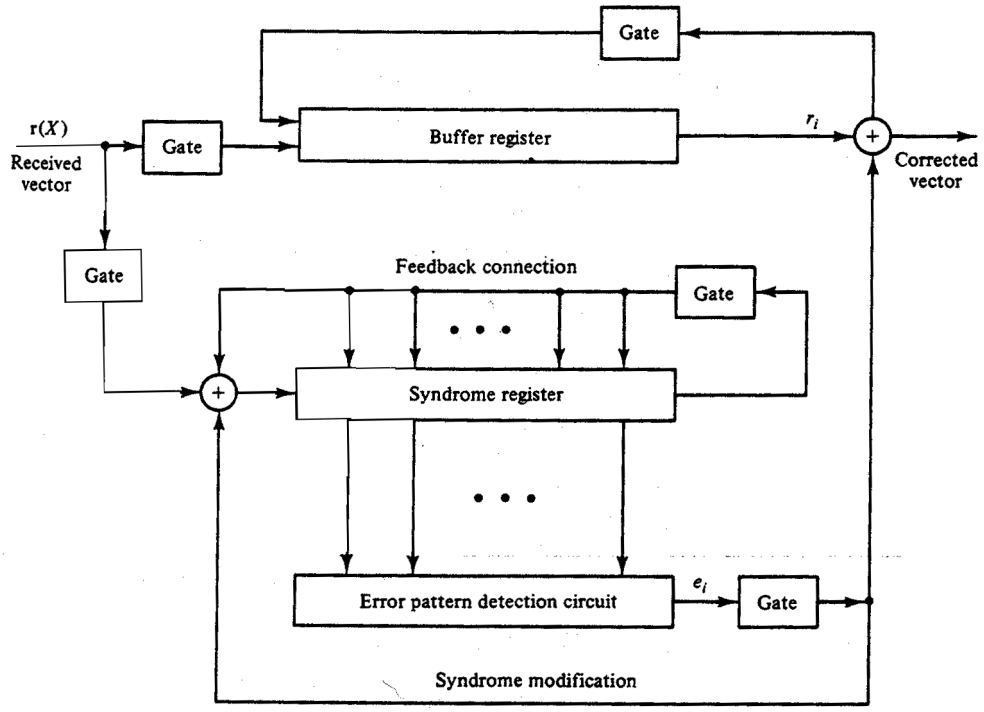
αποκωδικοποίηση σταματά μετά από n μετακινήσεις. Αν το e(x) είναι ένα διορθώσιμο error pattern τα περιεχόμενα του καταχωρητή του συνδρόμου θα είναι μηδέν στο τέλος της διαδικασίας αποκωδικοποίησης και το λαμβανόμενο διάνυσμα r (X) θα έχει αποκωδικοποιηθεί σωστά. Αν ο καταχωρητής του συνδρόμου δεν είναι μηδέν στο τέλος της διαδικασίας αποκωδικοποίησης ένα μη διορθώσιμο λάθος έχει ανιχνευθεί.

Ένας γενικός αποκωδικοποιητής για ένα (n,k) κυκλικό κώδικα φαίνεται στο σχήμα 4.4 και αποτελείται από τρία βασικά μέρη : (1) ένα καταχωρητή συνδρόμου (2) ένα ανιχνευτή για error pattern (3) ένα καταχωρητή buffer που χρησιμοποιείται για την αποθήκευση του λαμβανόμενου διανύσματος. Το λαμβανόμενο πολυώνυμο μεταφέρεται στο καταχωρητή συνδρόμου από το αριστερό άκρο. Για να απαλλάξουμε το σύνδρομο από ένα λάθος ψηφίο απλώς οδηγούμε το λάθος ψηφίο καταχωρητή ολίσθησης από τα αριστερά και μέσω μιας EXCLUSIVE – OR πύλης .

Η διαδικασία αποκωδικοποίησης περιγράφεται συνοπτικά ως εξής :

Βήμα 1 Το σύνδρομο διαμορφώνεται με τη μεταφορά ολόκληρου του λαμβανόμενου διανύσματος στο καταχωρητή συνδρόμου. Την ίδια στιγμή το λαμβανόμενο διάνυσμα αποθηκεύεται σε ένα buffer καταχωρητή.

Βήμα 2 Το σύνδρομο διαβάζεται από τον ανιχνευτή και ελέγχεται για το αντίστοιχο error pattern. Ο ανιχνευτής είναι ένα συνδυαστικό λογικό κύκλωμα το οποίο σχεδιάζεται με τέτοιο τρόπο έτσι ώστε η έξοδος του να είναι 1 αν και μόνο αν το σύνδρομο που βρίσκεται στον καταχωρητή αντιστοιχεί σε ένα διορθώσιμο error pattern με το λάθος να βρίσκεται στην υψηλότερης τάξης θέση τη X^{n-1} . Αυτό σημαίνει ότι αν ένας 1 εμφανίζεται στην έξοδο του ανιχνευτή, το λαμβανόμενο ψηφίο που βρίσκεται στην πιο δεξιά θέση στον buffer καταχωρητή θεωρείται ότι είναι λανθασμένο και ότι πρέπει να διορθωθεί, αν ένα 0 εμφανιστεί στην έξοδο του ανιχνευτή τότε το λαμβανόμενο ψηφίο που βρίσκεται στην πιο δεξιά θέση στον buffer καταχωρητή θεωρείται ότι είναι σωστό και καμιά διόρθωση δεν είναι αναγκαία. Άρα η έξοδος του ανιχνευτή είναι η εκτιμώμενη λανθασμένη τιμή του συμβόλου που βγαίνει από τον buffer.



Σχήμα 4.4 Αποκωδικοποιητής κυκλικού κώδικα με το λαμβανόμενο πολυώνυμο να ολισθαίνει στον καταχωρητή συνδρόμου από το αριστερό άκρο.

Βήμα 3 Το πρώτο λαμβανόμενο σύμβολο διαβάζεται από τον buffer. Την ίδια στιγμή ο καταχωρητής συνδρόμου μεταφέρεται μια φορά. Αν το πρώτο λαμβανόμενο σύμβολο ανιχνεύεται ως λανθασμένο σύμβολο διορθώνεται στην έξοδο του ανιχνευτή. Η έξοδος του ανιχνευτή επίσης οδηγείται πίσω στον καταχωρητή συνδρόμου για να τροποποιήσει το σύνδρομο(να εξαλείψει το λάθος από το σύνδρομο). Αυτό έχει ως αποτέλεσμα ένα νέο σύνδρομο που αντιστοιχεί στο τροποποιημένο λαμβανόμενο διάνυσμα που έχει μετακινηθεί μια φορά προς τα δεξιά.

Βήμα 4 Το νέο σύνδρομο που διαμορφώνεται στο βήμα 3 χρησιμοποιείται για να ανιχνεύσει αν το δεύτερο λαμβανόμενο σύμβολο(αυτό που βρίσκεται στην πιο δεξιά θέση στο buffer) είναι λανθασμένο. Ο αποκωδικοποιητής επαναλαμβάνει τα βήματα 2 και 3.Το δεύτερο λαμβανόμενο σύμβολο διορθώνεται με τον ίδιο ακριβώς τρόπο που διορθώθηκε και το πρώτο.

Βήμα 5 Ο αποκωδικοποιητής αποκωδικοποιεί το λαμβανόμενο διάνυσμα σύμβολο με τον τρόπο που περιγράφτηκε παραπάνω μέχρις ότου ολόκληρο το λαμβανόμενο διάνυσμα διαβαστεί από τον buffer καταχωρητή.

Ο αποκωδικοποιητής που περιγράφτηκε παραπάνω είναι γνωστός ως αποκωδικοποιητής Meggitt και μπορεί να χρησιμοποιηθεί σε κυκλικούς κώδικες. Το αν είναι πρακτικός ή όχι εξαρτάται ολοκληρωτικά από το κύκλωμα ανίχνευσης του error pattern.

Στο σημείο αυτό πρέπει να παρατηρήσουμε ότι ένας κυκλικός κώδικας αποκωδικοποιείται από ένα πολύ πιο απλό κύκλωμα από ότι ένας γραμμικός κώδικας. Το τίμημα όμως για αυτό είναι η ταχύτητα (μεγαλύτερος χρόνος αποκωδικοποίησης). Γενικά όμως πρέπει να γνωρίζουμε ότι απλότητα κυκλώματος και η ταχύτητα δεν μπορούν να επιτευχθούν και οι δύο ταυτόχρονα(trade off).

4.6 Κυκλικοί Hamming κώδικες

Οι Hamming κώδικες που παρουσιάστηκαν στην ενότητα 3.7 μπορούν να έχουν και κυκλική δομή. Ένας κυκλικός Hamming κώδικας μήκους $2^m - 1$ με $m \geq 3$ παράγεται από ένα αρχικό (primitive) πολυώνυμο p(X) βαθμού m.

Για να δείξουμε ότι ο παραπάνω κώδικας είναι ένας Hamming κώδικας εξετάζουμε τον πίνακα ελέγχου ισοτιμίας σε συστηματική μορφή με μέθοδο που παρουσιάστηκε στην παράγραφο 4.2. Διαιρώντας το X^{m+i} με το γενεσιουργό (generator) πολυώνυμο p(X) για $0 \leq i \leq 2^m - m - 1$ παίρνουμε :

$$X^{m+i} = a_i(X)\, p(X) + b_i(X)$$

Όπου $b_i(X)$ έχει τη μορφή :

$$b_i(X) = b_{i0} + b_{i1} X + \ldots + b_{i,m-1} X^{m-1}$$

Από τη στιγμή που X δεν είναι παράγοντας του βασικού (primitive) πολυώνυμου p(X), X^{m+i} και p(X) πρέπει να είναι πρώτοι (prime) και αυτό έχει σαν αποτέλεσμα ότι $b_i(X) \neq 0$ και $b_i(X)$ αποτελείται από τουλάχιστον δύο όρους.(Απόδειξη &4.6)

Έστω $H = [I_m \; Q]$ ο πίνακας ελέγχου ισοτιμίας του κυκλικού κώδικα που παράγεται από το ρ(X)

(I_m = $m \times m$ μοναδιαίος πίνακας και ο Q $m \times (2^m - m - 1)$ πίνακας). Έστω $b_i = (b_{i0}, b_{i1}, \ldots, b_{i,m-1})$ είναι το m-tuple που αντιστοιχεί στο $b_i(X)$. Ο πίνακας Q έχει $(2^m - m - 1)$ b_i με $0 \leq i \leq 2^m - m - 1$ ως στήλες του πίνακα για τις οποίες ισχύει ότι δεν υπάρχουν δύο ίδιες στήλες και ότι κάθε στήλη

περιέχει τουλάχιστον δύο 1.Επομένως ο πίνακας Η είναι πράγματι ο πίνακας ελέγχου ισοτιμίας ενός Hamming κώδικα και το ρ(Χ) είναι το γενεσιουργό (generator) πολυώνυμο. Η διαδικασία αποκωδικοποίησης που ισχύει για ένα Hamming κώδικα περιγράφεται με τα παρακάτω βήματα:

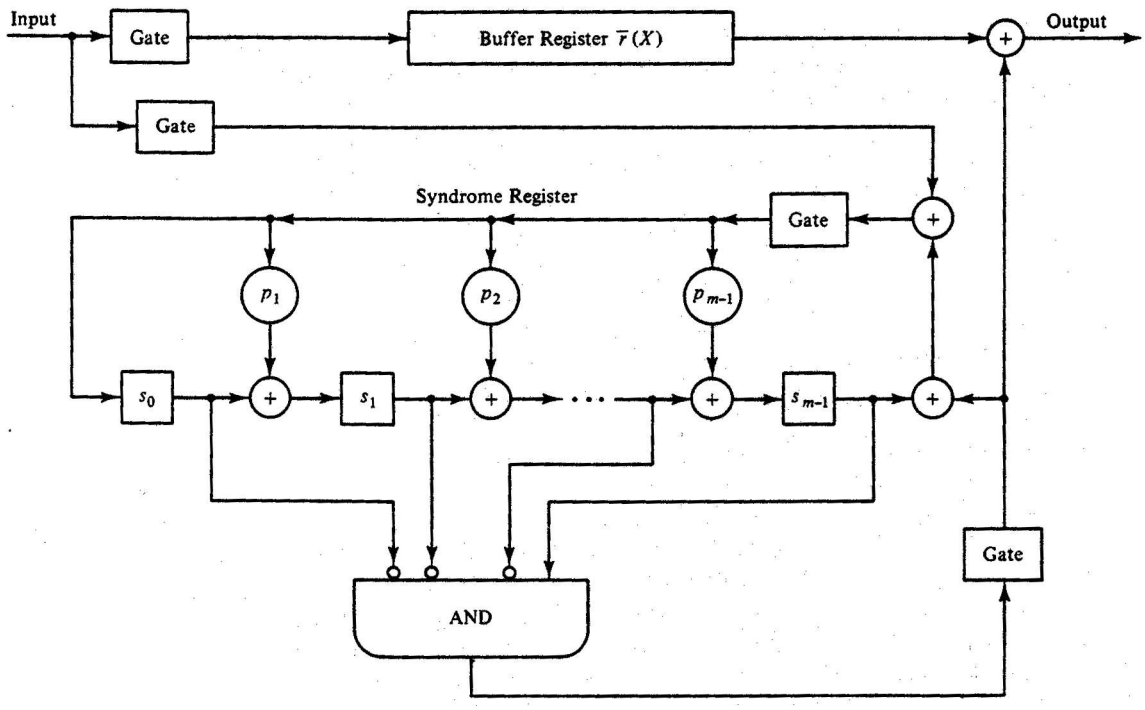
Βήμα 1 το σύνδρομο διαμορφώνεται με τη μεταφορά ολόκληρου του λαμβανόμενου διανύσματος στον καταχωρητή του συνδρόμου. Την ίδια στιγμή το λαμβανόμενο διάνυσμα αποθηκεύεται στον buffer καταχωρητή. Αν το σύνδρομο είναι μηδέν τότε ο αποκωδικοποιητής θεωρεί ότι δεν έγινε κανένα λάθος και καμιά διόρθωση δεν είναι αναγκαία. Αν το σύνδρομο δεν είναι μηδέν τότε ο αποκωδικοποιητής θεωρεί ότι ένα απλό λάθος έγινε.

Βήμα 2 Η λαμβανόμενη λέξη διαβάζεται από τον buffer καταχωρητή ψηφίο , ψηφίο. Κάθε φορά που διαβάζεται ένα ψηφίο από τον buffer καταχωρητή ο καταχωρητής του συνδρόμου μετακινείται μία φορά κυκλικά. Μόλις το σύνδρομο πάρει τη μορφή (0,0,0,...,1) το επόμενο ψηφίο που βγαίνει από τον buffer είναι το λανθασμένο ψηφίο και η έξοδος από την m εισόδων AND πύλη είναι 1.

Βήμα 3 Το λανθασμένο ψηφίο διαβάζεται από τον buffer καταχωρητή και διορθώνεται από την έξοδο της m εισόδων AND πύλης. Η διόρθωση γίνεται με μια πύλη EXCLUSIVE – OR.

Βήμα 4 Τα περιεχόμενα του καταχωρητή συνδρόμου μηδενίζονται μετά το διάβασμα ολόκληρου του λαμβανόμενου διανύσματος από τον buffer.

Το κύκλωμα αποκωδικοποιητή για ένα κυκλικό κώδικα Hamming φαίνεται στο σχήμα 4.5



Σχήμα 4.5 Αποκωδικοποιητής για ένα κυκλικό Hamming κώδικα

Οι δυνατότητες του κώδικα αυτού όσον αφορά τον αριθμό λαθών που μπορεί να διορθώσει είναι περιορισμένες. Ο κώδικας αυτός μπορεί να διορθώσει ένα απλό λάθος και ταυτόχρονα να ανιχνεύσει δύο λάθη.

Έστω g(X) = (X+1)p(X) όπου p(X) ένα βασικό (primitive) πολυώνυμο βαθμού m. Από τη στιγμή που και το p(X) και το (X+1) διαιρούν το $X^{(2^m)-1}+1$ άρα θα είναι και τα δύο πρώτα μεταξύ τους, το g(X) πρέπει επίσης να διαιρεί το $X^{(2^m)-1}+1$. Ένας κυκλικός κώδικας Hamming που διορθώνει ένα απλό λάθος και ταυτόχρονα ανιχνεύει δύο λάθη και έχει μήκος 2^m-1 παράγεται από το g(X) = (X+1)p(X). Ο κώδικας έχει m +1 ψηφία ελέγχου ισοτιμίας. Θα δειχτεί ότι η ελάχιστη απόσταση του κώδικα αυτού είναι 4.

Συμβολίζουμε με C1 τον κυκλικό κώδικα Hamming που διορθώνει ένα απλό λάθος και με C2 ένα κυκλικό κώδικα που παράγεται από το g(X) = (X+1)p(X). Αφού το p(X) είναι παράγοντας του g(X)

ο C1 περιέχει τον C2. Στην πραγματικότητα ο C2 περιέχει όλα τα κωδικά διανύσματα του C1 που έχουν ζυγό βάρος και αυτό γιατί κανένα κωδικό πολυώνυμο του C1 με περιττό βάρος δεν έχει τον X +1 σαν παράγοντα. Επομένως ένα κωδικό πολυώνυμο του C1 με περιττό βάρος δεν διαιρείται με το g(X) = (X+1)p(X) και δεν είναι κωδικό πολυώνυμο του C2.Όμως ένα κωδικό πολυώνυμο του C1 με ζυγό βάρος έχει το X +1 ως παράγοντα για αυτό το λόγο διαιρείται με το g(X) = (X+1)p(X) και είναι επίσης ένα κωδικό πολυώνυμο του C2.Από τα παραπάνω εξάγουμε το συμπέρασμα ότι το ελάχιστο βάρος του C2 είναι τουλάχιστον 4.

Το επόμενο βήμα είναι να αποδείξουμε ότι το ελάχιστο βήμα του C2 είναι ακριβώς 4.Έστω i,j και κ είναι τρεις διακριτοί μη αρνητικοί ακέραιοι μικρότεροι από 2^m –1 τέτοιοι ώστε $X^i+X^j+X^k$ δεν διαιρείται με το ρ(X).Για παράδειγμα πρώτα επιλέγουμε i και j. Διαιρώντας το X^i+X^j με το ρ(X) έχουμε :

$X^i+X^j = a(X)p(X) +b(X)$

Όπου b(X) είναι το υπόλοιπο βαθμού m –1 ή μικρότερου. Αφού το X^i+X^j δεν διαιρείται ακριβώς με το ρ(X) το $b(X) \neq 0$. Τώρα επιλέγουμε έναν ακέραιο κ τέτοιο ώστε όταν το X^k διαιρείται με το ρ(X) το υπόλοιπο δεν είναι ίσο με b(X).Επομένως το $X^i+X^j+X^k$ δεν διαιρείται με το ρ(X). Διαιρώντας αυτό το πολυώνυμο με ρ(X) έχουμε :

$X^i+X^j+X^k = c(X)\,\rho(X) + d(X).$

Τώρα επιλέγουμε ένα μη αρνητικό ακέραιο l μικρότερο από 2^m –1 τέτοιο ώστε όταν το X^l διαιρείται με το ρ(X) το υπόλοιπο είναι το d(X) δηλαδή :

$X^l = f(X)p(X) + d(X)$

Ο ακέραιος l δεν μπορεί να είναι ίσος με κανένα από τους i,j και κ. Έστω ο l = i. Τότε θα είχαμε

$X^j+X^k = [c(X)+ f(X)]\,\rho(X)$

Αυτό σημαίνει ότι το ρ(X) διαιρεί το $X^{k-j}+1$ που αυτό είναι όμως αδύνατο αφού $k-j < 2^m$ –1 και το ρ(X) είναι βασικό (primitive) πολυώνυμο. Άρα θα ισχύει

$X^i+X^j+X^k + X^l = [c(X)+ f(X)]\,\rho(X)$

Αφού ο X +1 είναι παράγοντας του $X^i+X^j+X^k + X^l$ και δεν είναι παράγοντας του ρ(X) το c(X)+ f(X) πρέπει να διαιρείται με το X+1 και έτσι το $X^i+X^j+X^k + X^l$ διαιρείται με το g(X) = (X+1)p(X) κάτι που σημαίνει ότι είναι ένα κωδικό διάνυσμα ενός κώδικα που παράγεται από το g(X) και έχει βάρος 4. Αυτό αποδεικνύει ότι ο κυκλικός κώδικας C2 που παράγεται από το g(X) = (X+1)p(X)έχει ελάχιστη απόσταση 4 και για αυτό μπορεί να διορθώνει ένα απλό λάθος και ταυτόχρονα να ανιχνεύει κάθε συνδυασμό με δύο λάθη.

Το κύκλωμα αποκωδικοποίησης του προηγούμενου σχήματος για κώδικα Hamming που διορθώνει ένα απλό λάθος μπορεί να τροποποιηθεί σε ένα κύκλωμα για ένα κώδικα Hamming που διορθώνει ένα απλό λάθος και ανιχνεύει δύο λάθη ταυτόχρονα και αυτό φαίνεται στο σχήμα 4.6

Σχήμα 4.6 Κύκλωμα αποκωδικοποίησης για ένα κυκλικό Hamming κώδικα που ανιχνεύει ένα ή δύο λάθη.

Έστω r(X) είναι το λαμβανόμενο πολυώνυμο. Διαιρώντας το X^mr(X) με ρ(X) και το r(X) με το (X+1) θα έχουμε :

$X^m r(X) = a_1(X)p(X) + s_p(X)$

$r(X) = a_2(X)(X+1) + \sigma$

όπου $s_p(X)$ είναι βαθμού m-1 ή μικρότερου και σ είναι ή 0 ή 1. Αν $s_p(X) = 0$ και $\sigma = 0$ τότε το r(X) διαιρείται με το (X+1) p(X) και είναι κωδικό πολυώνυμο, διαφορετικά το r(X) δεν είναι κωδικό πολυώνυμο. Ορίζουμε το σύνδρομο του r(X) ως :

$s(X) = X\, s_p(X) + \sigma$

Αν ένα απλό λάθος συμβεί τότε το $s_p(X) \neq 0$ και το σ =1. Όταν συμβεί ένα error pattern με δύο λάθη τότε θα έχουμε $s_p(X) \neq 0$ και το σ =0.Σε όλα αυτά τα δεδομένα βασίζεται και το κύκλωμα αποκωδικοποίησης του σχήματος 4.6. Για τη διόρθωση λαθών και την ανίχνευση λαθών οι διαδικασίες που ακολουθούνται περιγράφονται ως εξής :

1. Για σ = 0 και $s_p(X) = 0$ ο αποκωδικοποιητής υποθέτει ότι δεν υπάρχει κανένα λάθος στο λαμβανόμενο πολυώνυμο.

2. Για σ = 1 και $s_p(X) \neq 0$, ο αποκωδικοποιητής υποθέτει ότι ένα απλό λάθος συνέβη και προχωρά σε μια διαδικασία διόρθωσης λαθών όπως αυτή που περιγράφτηκε στην αποκωδικοποίηση ενός απλού λάθους στο κυκλικό κώδικα Hamming.

3. Για σ =0 και $s_p(X) \neq 0$ ο αποκωδικοποιητής υποθέτει ότι δύο λάθη συνέβησαν και ανάβει ο συναγερμός λάθους.

4. Για σ = 1 και $s_p(X) = 0$ ο συναγερμός λάθους επίσης ανάβει. Αυτό συμβαίνει όταν ένα error pattern με περιττό αριθμό (μεγαλύτερο από 1) λαθών συμβεί και το error pattern διαιρείται με το ρ(X).

Η πιθανότητα Pu(E) για ένα μη ανιχνεύσιμο λάθος δίνεται από τον παρακάτω τύπο :

$$Pu(E) = 2^{-(m+1)} \{1 + 2(2^m-1)(1-p)(1-2p)^{2^{(m-1)}-1} + (1-2p)^{2^m-1}\} - (1-p)^{2^m-1}$$

Η παραπάνω σχέση δείχνει ότι ο κώδικας Hamming ικανοποιεί το άνω φράγμα

$$2^{-(n-k)} = 2^{-(m+1)}$$

4.7 Σύντομοι κώδικες Hamming

Υπάρχουν τεχνικές για τη δημιουργία πιο απλών και σύντομων κωδίκων. Αυτές οι τεχνικές αποσκοπούν σε ευκολότερες διαδικασίες κωδικοποίησης και αποκωδικοποίησης. Σε αυτό το σημείο παρουσιάζεται περιληπτικά μια από αυτές.

Δεδομένου ενός (n,k) κυκλικού κώδικα C θεωρούμε το σύνολο των κωδικών διανυσμάτων για τα οποία τα t ψηφία που οδηγούν τα κώδικα διανύσματα και βρίσκονται σε θέσεις υψηλής τάξης είναι μηδέν. Υπάρχουν $2^{\kappa - t}$ τέτοια διανύσματα τα οποία διαμορφώνουν ένα γραμμικό υποκώδικα του C. Αν τα t μηδενικά ψηφία πληροφορίας διαγραφούν από κάθε ένα από αυτά τα διανύσματα έχουμε ένα σύνολο διανυσμάτων μήκους n – t . Αυτά τα διανύσματα διαμορφώνουν ένα (n – t, κ – t) γραμμικό κώδικα που ονομάζεται συντομότερος (shortened) κυκλικός κώδικας και δεν είναι

κυκλικός. Ένας τέτοιος κώδικας έχει τις ίδιες ικανότητες διόρθωσης λαθών με τον κώδικα από τον οποίο προήλθε.

Η κωδικοποίηση και η αποκωδικοποίηση μπορεί να γίνει με την χρήση των ίδιων κυκλωμάτων που χρησιμοποιούνται για τους κυκλικούς κώδικες και αυτό γιατί η διαγραφή των tψηφίων δεν επηρεάζουν τους υπολογισμούς που γίνονται για το σύνδρομο και τον έλεγχο ισοτιμίας. Αυτό όμως που πρέπει να γίνει είναι μόλις το λαμβανόμενο διάνυσμα εισέλθει ολόκληρο στον καταχωρητή συνδρόμου ο καταχωρητής αυτός πρέπει να μετακινηθεί κυκλικά t φορές για να παράγει το σωστό σύνδρομο για την αποκωδικοποίηση του πρώτου λαμβανόμενου ψηφίου

r_{n-t-1}. Όπως είναι φυσικό μπορούμε να δούμε ότι για μεγάλο t οι επιπλέον t μετακινήσεις δημιουργούν μια μη επιθυμητή καθυστέρηση και αυτό μπορεί να εξαλειφθεί με το να αλλάξουμε ή τις συνδέσεις του καταχωρητή συνδρόμου ή τις συνδέσεις του κυκλώματος ανίχνευσης των error pattern.

Έστω $r(X) = r_0 + r_1X + \ldots + r_{n-l-1} X^{n-t-1}$ το λαμβανόμενο πολυώνυμο. Έστω ότι το πολυώνυμο αυτό μεταφέρεται στο καταχωρητή συνδρόμου από το δεξί άκρο. Αν χρησιμοποιηθεί το κύκλωμα αποκωδικοποίησης του αρχικού κυκλικού κώδικα στην αποκωδικοποίηση του συντομότερου κώδικα τότε το σωστό σύνδρομο για την αποκωδικοποίηση του λαμβανόμενου ψηφίου είναι ίσο με το υπόλοιπο της διαίρεσης του $X^{n-\kappa+t} r(X)$ με το πολυώνυμο g(X).

Έτσι όταν το πολυώνυμο r(X) μεταφέρεται στο καταχωρητή συνδρόμου από το δεξί άκρο αυτό ισοδυναμεί με το να πολλαπλασιάζεται αρχικά το r(X) με $X^{n-\kappa}$, μετά από τις t επιπλέον μετακινήσεις ολόκληρου του r(X) στον καταχωρητή. Αν θέλουμε να αποφύγουμε αυτές τις μετακινήσεις εργαζόμαστε ως εξής : Αρχικά διαιρώντας το $X^{n-\kappa+t} r(X)$ με το πολυώνυμο g(X) παίρνουμε

$$X^{n-\kappa+t} r(X) = a_1(X)\, g(X) + s^{(n-\kappa+t)}(X)$$

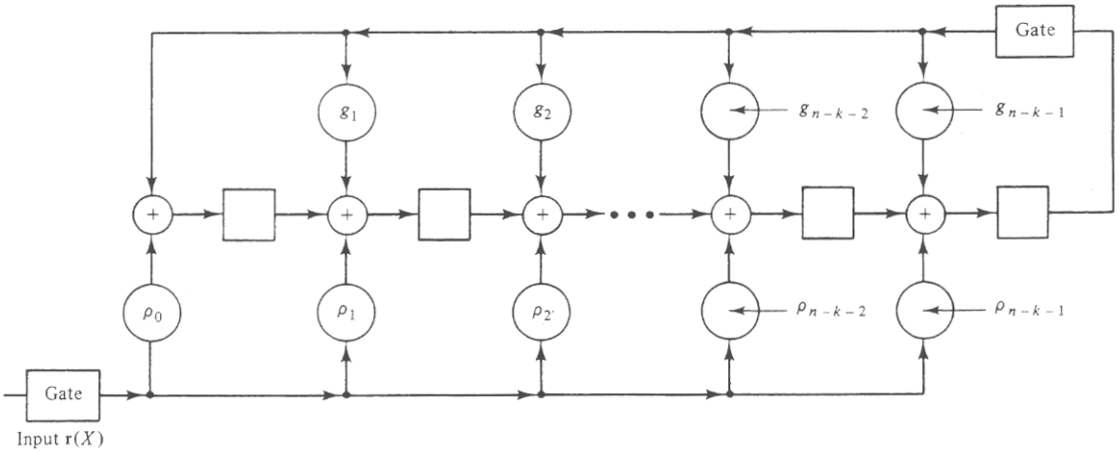
Όπου $s^{(n-\kappa+t)}(X)$ είναι το υπόλοιπο και είναι το σύνδρομο για την αποκωδικοποίηση του ψηφίου r_{n-t-1}. Μετά διαιρούμε το $X^{n-\kappa+t}$ με το g(X).Έστω $\rho(X) = \rho_0 + \rho_1X + \ldots + \rho_{n-\kappa-l} X^{n-\kappa-1}$ είναι το υπόλοιπο αυτής της διαίρεσης. Τότε προκύπτει η παρακάτω σχέση

$$\rho(X) = X^{n-\kappa+t} + a_2(X)\, g(X)$$

πολλαπλασιάζοντας και τα δύο μέλη της παραπάνω σχέσης και με αντικατάσταση έχουμε

$$\rho(X)\, r(X) = [a_1(X) + a_2(X)\, r(X)]\; g(X) + s^{(n-\kappa+t)}(X)$$

Η παραπάνω ισότητα μας δείχνει ότι το σύνδρομο $s^{(n-\kappa+t)}(X)$ το παίρνουμε όταν πολλαπλασιάζουμε το $\rho(X)$ με το $r(X)$ και μετά διαιρούμε αυτό το γινόμενο με $g(X)$ και με αυτό τον τρόπο αποφεύγονται οι επιπλέον μετακινήσεις. Αυτή η υλοποίηση φαίνεται στο σχήμα 4.7



Σχήμα 4.7 Κύκλωμα πολλαπλασιασμού του $r(X)$ με το $\rho(X) = \rho_0 + \rho_1 X + \ldots + \rho_{n-\kappa-1} X^{n-\kappa-1}$ και διαίρεσης του $\rho(X)r(X)$ με το $g(X) = 1 + g_1 X + g_2 X^2 + \ldots + g_{n-k-1} X^{n-k-1} + X^{n-k}$.

Κεφάλαιο 5 - Αποκωδικοποίηση παγίδευσης λάθους για κυκλικούς κώδικες

Στο κεφάλαιο αυτό παρουσιάζεται μία διαφοροποίηση της αποκωδικοποίησης Meggitt που ονομάζεται αποκωδικοποίηση παγίδευσης λάθους(error trapping decoding). Ένας αποκωδικοποιητής που βασίζεται σε αυτήν την τεχνική εφαρμόζεται σε ένα απλό λογικό συνδυαστικό κύκλωμα για την ανίχνευση και διόρθωση λαθών. Χρησιμοποιείται αποτελεσματικά στις περιπτώσεις (1) κώδικες διόρθωσης απλού λάθους (2) σε μερικούς σύντομους κώδικες διόρθωσης δύο λαθών (3) σε κώδικες διόρθωσης λαθών σε μορφή « ξεσπάσματος ». Δεν είναι αποτελεσματικός σε περιπτώσεις κωδίκων που είναι μεγάλοι και έχουν υψηλές συχνότητες καθώς επίσης σε περιπτώσεις που απαιτείται διόρθωση πολλών λαθών (στην τελευταία περίπτωση είναι δυνατόν να γίνουν αρκετές βελτιώσεις όπως θα δούμε στο κεφάλαιο αυτό).

5.1 Αποκωδικοποίηση παγίδευσης λαθών

Έστω ένας κυκλικός κώδικας (n, k) με γενεσιουργό (generator) πολυώνυμο g(X). Υποθέτουμε ότι ένα κωδικό διάνυσμα v(X) μεταδίδεται και αλλοιώνεται από ένα error pattern e(X).Τότε το λαμβανόμενο πολυώνυμο θα είναι r(X)= v(X) + e(X). Το σύνδρομο s(X) που υπολογίζεται από το r(X) θα είναι ίσο με το υπόλοιπο της διαίρεσης του error pattern e(X) με το g(X).δηλαδή

$e(X) = \alpha(X)\, g(X) + s(X)$

Υποθέτουμε ότι τα λάθη βρίσκονται περιορισμένα στις n – k υψηλής τάξης θέσεις $X^{\kappa}, X^{\kappa+1},...,$

X^{n-1} του r(X). Αν το r(X) μετακινηθεί κυκλικά n – k φορές τα λάθη θα βρεθούν περιορισμένα στις n – k χαμηλής τάξης θέσεις $X^{0}, X^{1},..., X^{n-\kappa-1}$ του $r^{(n-k)}(X)$. Το αντίστοιχο error pattern θα είναι

$e^{(n-\kappa)}(X) = e_{\kappa} + e_{\kappa+1} X + ... + e_{n-1} X^{n-\kappa-1}$

Από τη στιγμή που το σύνδρομο $s^{(n-k)}(X)$ του $r^{(n-k)}(X)$ ισούται με το υπόλοιπο της διαίρεσης του

$e^{(n-k)}(X)$ με το g(X) και από τη στιγμή που ο βαθμός του $e^{(n-k)}(X)$ είναι μικρότερος από n – k έχουμε την παρακάτω ισότητα :

$s^{(n-k)}(X) = e^{(n-k)}(X) = e_\kappa + e_{\kappa+1} X + ... + e_{n-1} X^{n-\kappa-1}$

Πολλαπλασιάζοντας το $s^{(n-k)}(X)$ με το X^κ θα έχουμε

$X^\kappa s^{(n-k)}(X) = e(X) = e_\kappa X^\kappa + e_{\kappa+1} X^{\kappa+1} + ... + e_{n-1} X^{n-1}$

Αυτό σημαίνει ότι αν τα λάθη βρίσκονται περιορισμένα στις n – k υψηλής τάξης θέσεις του λαμβανόμενου πολυώνυμου r(X), το error pattern e(X) είναι ίσο με $X^\kappa s^{(n-k)}(X)$ όπου $s^{(n-k)}(X)$ είναι το σύνδρομο του $r^{(n-k)}(X)$ δηλαδή η (n – k) κυκλική μετακίνηση του r(X). Το διάνυσμα που προκύπτει είναι το μεταδιδόμενο κωδικό διάνυσμα. Γενικά υποθέτουμε ότι τα λάθη βρίσκονται περιορισμένα σε n – k συνεχείς θέσεις $X^i, X^{i+1}, ..., X^{(n-k)+i-1}$ του r(X). Αν το r(X) μετακινηθεί κυκλικά n – i φορές προς τα δεξιά τα λάθη θα βρεθούν περιορισμένα στις n – k χαμηλής τάξης θέσεις $X^0, X^1, ..., X^{n-\kappa-1}$ του $r^{(n-i)}(X)$ και το error pattern θα είναι ίσο με είναι ίσο με

$X^i s^{(n-i)}(X)$ όπου το $s^{(n-i)}(X)$ είναι το σύνδρομο του $r^{(n-i)}(X)$.

Γενικά η μεταφορά (shifting) του καταχωρητή συνδρόμου μέχρι ότου τα περιεχόμενα γίνουν ίδια με τα λανθασμένα ψηφία ονομάζεται παγίδευση λάθους (error trapping). Αν τα λάθη βρίσκονται περιορισμένα σε n – k συνεχόμενες θέσεις του r(X) και μπορούμε να ανιχνεύσουμε πότε τα λάθη παγιδεύονται στον καταχωρητή συνδρόμου, η διόρθωση λαθών μπορεί να γίνει απλά με το να προσθέσουμε τα περιεχόμενα του καταχωρητή συνδρόμου με τα λαμβανόμενα ψηφία στις n – k κατάλληλες θέσεις.

Υποθέτουμε ότι χρησιμοποιούμε ένα κυκλικό κώδικα για τη διόρθωση t λαθών. Για να ανιχνεύσουμε το γεγονός ότι τα λάθη έχουν παγιδευτεί στον καταχωρητή συνδρόμου μπορούμε απλώς να ελέγχουμε το βάρος του συνδρόμου μετά από κάθε μεταφορά του καταχωρητή του συνδρόμου. Μόλις το βάρος του συνδρόμου γίνει μικρότερο ή ίσο του t υποθέτουμε ότι τα λάθη έχουν παγιδευτεί στον καταχωρητή του συνδρόμου. Αν ο αριθμός των λαθών του r(X) είναι μικρότερος ίσος του t και αν αυτά είναι μαζεμένα στις n – k συνεχόμενες θέσεις , τα λάθη παγιδεύονται στον καταχωρητή του συνδρόμου μόνο όταν το βάρος του συνδρόμου στον καταχωρητή γίνει μικρότερο ίσο του t.

Αυτό αποδεικνύεται ως εξής :

Ένα error pattern e(X) με t ή λιγότερα λάθη που βρίσκονται μαζεμένα σε n – k συνεχόμενες θέσεις πρέπει να έχει τη μορφή $e(X) = X^j B(X)$ όπου B(X) έχει t ή λιγότερους όρους και είναι βαθμού

n – k – 1 ή μικρότερου. Διαιρώντας το e(X) με το γενεσιουργό (generator) πολυώνυμο g(X) έχουμε :

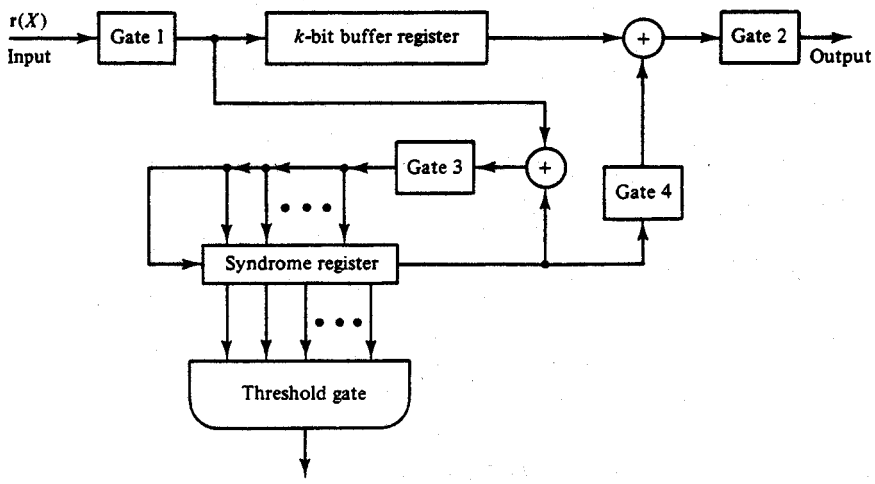
$X^j B(X) = a(X)g(X) + s(X)$

Όπου s (X) είναι το σύνδρομο του X^j B(X). Από τη στιγμή που το X^j B(X) + s(X) είναι πολλαπλάσιο του g(X) είναι ένα κωδικό πολυώνυμο. Το σύνδρομο s(X) δεν μπορεί να έχει βάρος t ή λιγότερο εκτός και αν $s(X) = X^j B(X)$.Υποθέτουμε ότι το βάρος του s(X) είναι t ή λιγότερο και $s(X) \neq X^j B(X)$. Τότε το $X^j B(X) + s(X)$ είναι ένα μη μηδενικό κωδικό διάνυσμα με βάρος λιγότερο από 2 t +1. Αυτό είναι αδύνατο από τη στιγμή που ένας κώδικας που διορθώνει t λάθη έχει ελάχιστο βάρος τουλάχιστον 2 t +1. Επομένως συμπεραίνουμε ότι τα λάθη είναι παγιδευμένα στον καταχωρητή του συνδρόμου μόνο όταν το βάρος του συνδρόμου γίνει t ή λιγότερο.

Η λειτουργία ενός αποκωδικοποιητή που βασίζεται στην μέθοδο αποκωδικοποίησης παγίδευσης λάθους φαίνεται στο σχήμα 5.1. Παρακάτω περιγράφεται βήμα προς βήμα όλες οι λειτουργίες ενός τέτοιου αποκωδικοποιητή

Βήμα 1 Το λαμβανόμενο πολυώνυμο r(X) μεταφέρεται στον buffer και ταυτόχρονα στον καταχωρητή του συνδρόμου με τις πύλες 1 και 3 ανοικτές και όλες τις άλλες πύλες κλειστές . Από τη στιγμή που ενδιαφερόμαστε μόνο για την ανάκτηση των κ ψηφίων πληροφορίας ο buffer έχει να αποθηκεύσει τα κ λαμβανόμενα ψηφία πληροφορίας.

Βήμα 2 Μόλις ολόκληρο το r(X) μπει στον καταχωρητή του συνδρόμου ελέγχεται το βάρος του συνδρόμου στον καταχωρητή από μία πύλη κατωφλίου (n – k) εισόδων της οποίας η έξοδος είναι 1 όταν t ή λιγότερες από τις εισόδους είναι 1 διαφορετικά είναι 0. Α) αν το βάρος του συνδρόμου είναι t ή λιγότερο, τα ψηφία συνδρόμου που βρίσκονται στον καταχωρητή συνδρόμου είναι ίδια με τα ψηφία λάθους που βρίσκονται στις n – k υψηλότερες θέσεις του X^κ , $X^{\kappa+1}$,..., X^{n-1} του r(X). Τότε οι πύλες 2 και 4 ανοίγουν και κλείνουν όλες οι υπόλοιπες. Το λαμβανόμενο διάνυσμα διαβάζεται από τον buffer καταχωρητή ένα ψηφίο κάθε φορά και διορθώνεται από τα λανθασμένα ψηφία που βγαίνουν από τον καταχωρητή του συνδρόμου. Β) Αν το βάρος του συνδρόμου είναι μεγαλύτερο από t, τότε τα λάθη δεν είναι μαζεμένα στις n – k υψηλότερες θέσεις του r(X) και δεν έχουν παγιδευτεί στον καταχωρητή του συνδρόμου οπότε πάμε στο βήμα 3.



Σχήμα 5.1 Αποκωδικοποιητής παγίδευσης λαθών

Βήμα 3 Μεταφέρουμε κυκλικά τον καταχωρητή του συνδρόμου μία φορά με την πύλη 3 ανοικτή και τις υπόλοιπες κλειστές. Το βάρος του νέου συνδρόμου ελέγχεται. Α) Αν αυτό είναι t ή λιγότερο, τα λάθη είναι μαζεμένα στις θέσεις $X^{\kappa-1}$, X^{κ} ,..., X^{n-2} του r(X) και τα περιεχόμενα του καταχωρητή του συνδρόμου είναι ίδια με τα λάθη σε αυτές τις θέσεις. Αφού το πρώτο ψηφίο r_{n-1} δεν είναι λανθασμένο διαβάζεται από τον buffer καταχωρητή με την πύλη 2 ανοικτή. Μόλις διαβαστεί το r_{n-1} ανοίγει η πύλη 4 και κλείνει η πύλη 3. Τα περιεχόμενα του καταχωρητή του συνδρόμου μεταφέρονται έξω και χρησιμοποιούνται για την διόρθωση των n – k λαμβανόμενων ψηφίων που βγαίνουν από τον buffer καταχωρητή. Β) Αν το βάρος του συνδρόμου είναι μεγαλύτερο από t, τότε μεταφέρουμε κυκλικά τον καταχωρητή του συνδρόμου μία ακόμη φορά με την πύλη 3 ανοικτή.

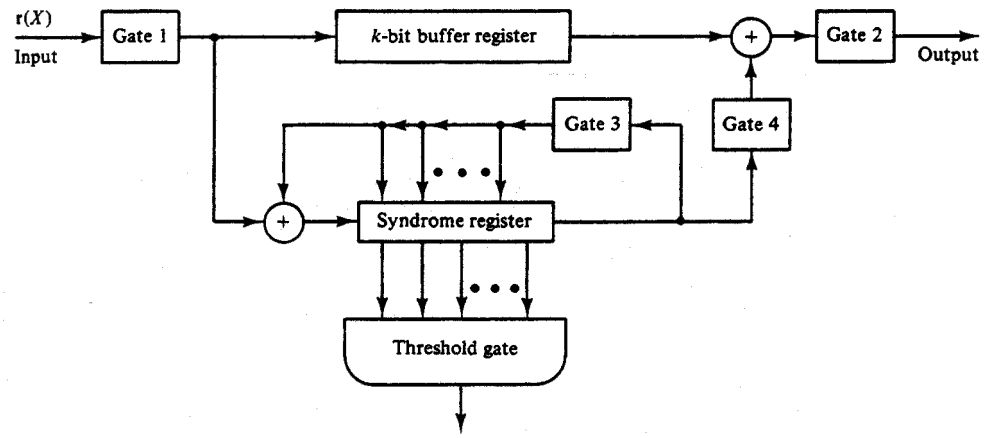
Βήμα 4 Ο καταχωρητής του συνδρόμου συνεχώς μεταφέρεται μέχρι ότου το βάρος των περιεχομένων του να γίνει t ή λιγότερο. Αν το βάρος γίνει t ή λιγότερο σο τέλος της ith μεταφοράς με $1 \leq i \leq \kappa$ τα πρώτα λαμβανόμενα ψηφία r_{n-i}, r_{n-i+1},..., r_{n-1} στο buffer καταχωρητή είναι αποδεσμευμένα από λάθη και τα περιεχόμενα του καταχωρητή του συνδρόμου είναι ίδια με τα λάθη στις θέσεις $X^{\kappa-i}$, $X^{\kappa-i+1}$,..., X^{n-i-1} . Μόλις διαβαστεί το i λαμβανόμενο ψηφίο που δεν έχει λάθος από τον buffer καταχωρητή τα περιεχόμενα του καταχωρητή του συνδρόμου μεταφέρονται εκτός και χρησιμοποιούνται για τη διόρθωση του των επόμενων n – k λαμβανόμενων ψηφίων που βγαίνουν από τον buffer καταχωρητή. Όταν τα κ λαμβανόμενα ψηφία πληροφορίας διαβαστούν από τον buffer καταχωρητή και διορθωθούν η πύλη 2 κλείνει. Κάθε μη μηδενικό στοιχείο που μένει στον καταχωρητή του συνδρόμου είναι λάθος του μέρους ισοτιμίας του r(X) και αγνοείται.

Βήμα 5 Αν το βάρος του συνδρόμου δεν γίνεται t ή λιγότερο την ώρα που ο καταχωρητής του συνδρόμου έχει μετακινηθεί κ φορές τότε είτε έχει συμβεί ένα error pattern με λάθη συνεχόμενα στις n – k συνεχόμενες ακραίες θέσεις είτε έχει συμβεί ένα μη διορθώσιμο error pattern. Συνεχίζουμε τη μετακίνηση του καταχωρητή του συνδρόμου. Υποθέτουμε ότι το βάρος των περιεχομένων γίνεται t ή λιγότερο στο τέλος των k + l μετακινήσεων όπου $1 \le l \le n - k$. Τότε τα λάθη περιορίζονται στις n – k συνεχόμενες ακραίες θέσεις $X^{\kappa-l}$, $X^{\kappa-l+1}$,..., X^{n-1} ,… X^{0} , X^{1} ,...,

$X^{n-k-l-1}$ του r(X) . Τα l ψηφία των l αριστερότερων θέσεων του καταχωρητή του συνδρόμου αντιστοιχούν στα λάθη στις l υψηλότερης τάξης θέσεις $X^{\kappa-l}$, $X^{\kappa-l+1}$,..., X^{n-1} του r(X). Από τη στιγμή που δεν χρειαζόμαστε τα λάθη στις n – k – l θέσεις ισοτιμίας μετακινούμε τον καταχωρητή του συνδρόμου n – k – l φορές με όλες τις πύλες κλειστές. Τώρα τα l λάθη των θέσεων X^{n-l} , X^{n-l+1} ,..., X^{n-1} του r(X) περιέχονται στις l δεξιότερες θέσεις του καταχωρητή του συνδρόμου. Με τις πύλες 2 και 4 ανοικτές και τις άλλες κλειστές τα λαμβανόμενα ψηφία στον buffer καταχωρητή διαβάζονται και διορθώνονται από τα αντίστοιχα λανθασμένα ψηφία που βγαίνουν από τον καταχωρητή του συνδρόμου. Έτσι τελειώνει και η διαδικασία αποκωδικοποίησης.

Αν το βάρος του συνδρόμου δεν γίνεται ποτέ t ή λιγότερο τη στιγμή που ο καταχωρητής του συνδρόμου έχει μετακινηθεί n φορές συνολικά είτε ένα μη διορθώσιμο error pattern έχει συμβεί είτε τα λάθη δεν βρίσκονται περιορισμένα σε n – k συνεχόμενες θέσεις. Σε κάθε περίπτωση τα λάθη ανιχνεύονται. Εκτός και αν τα λάθη βρίσκονται μαζεμένα σε n – k συνεχόμενες ακραίες θέσεις (και στα δύο άκρα του λαμβανόμενου διανύσματος) του r(X), τα ,λαμβανόμενα ψηφία πληροφορίας μπορούν να διαβαστούν από τον buffer καταχωρητή, να διορθωθούν και να παραδοθούν στα δεδομένα μετά από κ κυκλικές μετακινήσεις του καταχωρητή του συνδρόμου. Για μεγάλο n και n – k ο αριθμός των διορθώσιμων error patterns με λάθη που βρίσκονται τις ακραίες θέσεις είναι μεγάλος και προκαλεί μη επιθυμητή καθυστέρηση στην αποκωδικοποίηση.

Για αυτό το λόγο είναι επιθυμητή η εύρεση ενός άλλου, πιο γρήγορου τρόπου αποκωδικοποίησης των error pattern που έχουν τα λάθη στις n – k συνεχόμενες ακραίες θέσεις. Αυτό μπορεί να επιτευχθεί με την είσοδο του λαμβανόμενου διανύσματος r(X) στον καταχωρητή του συνδρόμου από το αριστερό άκρο όπως φαίνεται στο σχήμα 5.2



Σχήμα 5.2 Αποκωδικοποιητής παγίδευσης λαθών

Αυτή η διαφοροποίηση βασίζεται στο ότι αν τα λάθη είναι μαζεμένα στις $n - k$ χαμηλής τάξης θέσεις ισοτιμίας X^0 , X^1 ,..., X^{n-k-1} του r(X) τότε μετά την είσοδο ολόκληρου του r(X) στον καταχωρητή του συνδρόμου, τα περιεχόμενα του καταχωρητή είναι αυτά που αντιστοιχούν στα λάθος ψηφία στις θέσεις X^0 , X^1 ,..., X^{n-k-1} του r(X).Υποθέτουμε ότι τα λάθη δεν βρίσκονται στις $n - k$ χαμηλής τάξης θέσεις του r(X) αλλά στις $n - k$ συνεχόμενες θέσεις X^i , X^{i+1} ,..., $X^{n-k+i-1}$. Μετά από $n - i$ κυκλικές μετακινήσεις του r(X) τα λάθη θα μετακινηθούν στις $n - k$ χαμηλής τάξης θέσεις του $r^{(n-i)}(X)$ και το σύνδρομο του $r^{(n-i)}(X)$ θα είναι το ίδιο με τα λάθη που βρίσκονται μαζεμένα στις θέσεις X^i , X^{i+1} ,..., $X^{n-k+i-1}$ του r(X). Η λειτουργία του αποκωδικοποιητή που φαίνεται στο σχήμα 5.2 περιγράφεται βήμα προς βήμα ως εξής :

Βήμα 1 Πύλες 1 και 3 ανοικτές και οι άλλες πύλες κλειστές. Το λαμβανόμενο διάνυσμα r(X) μεταφέρεται στον καταχωρητή του συνδρόμου και στον buffer καταχωρητή(μόνο τα κ λαμβανόμενα ψηφία πληροφορίας) ταυτόχρονα. Μόλις ολόκληρο το r(X) μεταφερθεί στον καταχωρητή του συνδρόμου τα περιεχόμενα του διαμορφώνουν το σύνδρομο s(X) του r(X).

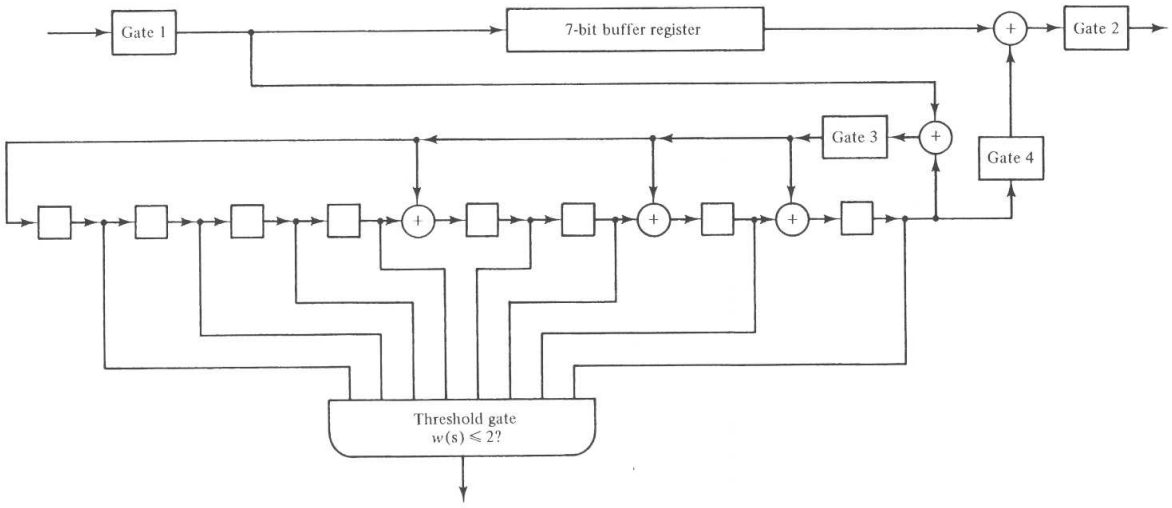
Βήμα 2 Ελέγχεται το βάρος του συνδρόμου. Α) Αν το βάρος είναι μικρότερο από t τα λάθη είναι περιορισμένα στις $n - k$ χαμηλής τάξης θέσεις ισοτιμίας X^0 , X^1 ,..., X^{n-k-1} του r(X). Επομένως τα κ λαμβανόμενα ψηφία πληροφορίας του buffer είναι σωστά. Η πύλη 2 ανοίγει και τα σωστά ψηφία πληροφορίας διαβάζονται από τον buffer με την πύλη 4 κλειστή. Β) Αν το βάρος είναι μεγαλύτερο από t ο καταχωρητής του συνδρόμου μετακινείται μία φορά με την πύλη 3 ανοικτή και τις άλλες πύλες κλειστές και πάμε στο βήμα 3.

Βήμα 3 Ελέγχεται το βάρος των νέων περιεχομένων του καταχωρητή του συνδρόμου. α) Αν το βάρος είναι μικρότερο ή ίσο του t, τότε τα λάθη βρίσκονται μαζεμένα στις θέσεις X^{n-1}, X^0,X^1, ..., X^{n-k-2} του r(x). Το πρώτο αριστερό ψηφίο του καταχωρητή συνδρόμου αντιστοιχεί στο λάθος της θέσης X^{n-1} του r(x) και τα υπόλοιπα n – k – 1 ψηφία αντιστοιχούν στα λάθη των θέσεων X^0,X^1, ..., X^{n-k-2} του r(x). Η έξοδος της πύλης κατωφλίου (threshold gate) απενεργοποιεί την πύλη 3 και θέτει σε λειτουργία ένα ρολόι που ξεκινά μέτρηση από το 2. Τότε ο καταχωρητής συνδρόμου μετακινείται με την πύλη 3 απενεργοποιημένη. Όταν η ένδειξη του ρολογιού γίνει n – k τα περιεχόμενα του καταχωρητή συνδρόμου θα είναι (0,0,0....1) όπου το 1 θα είναι το λάθος της θέσης X^{n-1} του r(x). Τα κ λαμβανόμενα ψηφία πληροφορίας τότε διαβάζονται από τον καταχωρητή και διορθώνεται το 1 καθώς βγαίνει από τον καταχωρητή συνδρόμου. Σε αυτή τη φάση η αποκωδικοποίηση ολοκληρώνεται. β) Αν το βάρος των περιεχομένων του συνδρόμου είναι μεγαλύτερο από t τότε ο καταχωρητής συνδρόμου μετακινείται μία ακόμα φορά με την πύλη 3 ανοικτή και τις υπόλοιπες κλειστές. Στη συνέχεια πάμε στο βήμα 4.

Βήμα 4 Το βήμα 3β επαναλαμβάνεται έως ότου τα περιεχόμενα του καταχωρητή συνδρόμου αποκτήσουν βάρος που είναι μικρότερο ή ίσο του t. Όταν το βάρος πέσει κάτω από t μετά την ith μετακίνηση για $1 \leq i \leq n - k$ το ρολόι ξεκινά τη μέτρηση από το i +1. Την ίδια στιγμή ο καταχωρητής συνδρόμου μετακινείται μία φορά με την πύλη 3 απενεργοποιημένη. Μόλις το ρολόι μετρήσει n – k τα δεξιά i ψηφία του καταχωρητή συνδρόμου αντιστοιχούν στα πρώτα i λαμβανόμενα ψηφία πληροφορίας του buffer καταχωρητή. Τα υπόλοιπα ψηφία πληροφορίας είναι όλα σωστά. Σε αυτό το σημείο οι πύλες 2 και 4 ενεργοποιούνται και τα λαμβανόμενα ψηφία πληροφορίας διαβάζονται από τον buffer για διόρθωση.

Βήμα 5 Αν το βάρος των περιεχομένων δεν μειώνεται σε t ενώ ο καταχωρητής έχει μετακινηθεί n – k φορές (με την πύλη 3 ανοικτή) η πύλη 2 ενεργοποιείται και τα λαμβανόμενα ψηφία πληροφορίας διαβάζονται από τον buffer ένα κάθε φορά. Την ίδια στιγμή ο καταχωρητής συνδρόμου μετακινείται με την πύλη 3 ανοικτή. Μόλις το βάρος των περιεχομένων του συνδρόμου μειωθεί σε t ή λιγότερο τα περιεχόμενα αντιστοιχούν στα λάθη των επόμενων n – k ψηφίων που θα εξέλθουν από τον buffer. Η πύλη 4 ανοίγει και τα λανθασμένα ψηφία πληροφορίας διορθώνονται από τα ψηφία που εξέρχονται από τον καταχωρητή συνδρόμου με την πύλη 3 κλειστή. Η πύλη 2 κλείνει μόλις τα κ ψηφία πληροφορίας διαβαστούν από τον buffer.

Στο σχήμα 5.3 παρουσιάζεται ένας αποκωδικοποιητής παγίδευσης λαθών για τον (15,7)BCH κώδικα που δημιουργείται από το $g(X) = 1+X^4+ X^6+ X^7+ X^8$.



Σχήμα 5.3 Αποκωδικοποιητής παγίδευσης λαθών για ένα κυκλικό κώδικα (15,7)

5.2 Βελτιωμένη αποκωδικοποίηση παγίδευσης λαθών

Η παραπάνω μέθοδος αποκωδικοποίησης μπορεί να βελτιωθεί δηλαδή να διορθώνει λανθασμένα patterns στα οποία τα περισσότερα λάθη βρίσκονται συγκεντρωμένα στις n – k συνεχόμενες θέσεις και κάποια λίγα λάθη βρίσκονται διασκορπισμένα έξω από αυτές τις θέσεις. Για αυτή τη βελτίωση φυσικά θα χρειαστεί επιπλέον εξοπλισμός η πολυπλοκότητα του οποίου εξαρτάται από τον αριθμό των λαθών που βρίσκονται έξω από τις n – k συνεχόμενες θέσεις και πρόκειται να διορθωθούν. Στη συνέχεια παρουσιάζεται μια βελτιωμένη έκδοση αυτής της αποκωδικοποίησης που προτείνεται από τον Kasami.

Το λανθασμένο pattern $e(X) = e_0 + e_1 X + ... + e_{n-\kappa-1} X^{n-\kappa-1}$ το οποίο αλλοίωσε το μεταδιδόμενο κωδικό διάνυσμα μπορεί να χωριστεί σε δύο μέρη :

$$e_p(X) = e_0 + e_1 X + ... + e_{n-k-1} X^{n-\kappa-1}$$

$$e_i(X) = e_{n-k} X^{n-k} + ... + e_{n-1} X^{n-1}$$

όπου το $e_i(X)$ περιέχει τα λάθη του τμήματος πληροφορίας του λαμβανόμενου διανύσματος και

$e_p(X)$ περιέχει τα λάθη του τμήματος ισοτιμίας του λαμβανόμενου διανύσματος. Διαιρώντας το $e_i(X)$ με το κωδικό γενεσιουργό (generator) πολυώνυμο g(X) έχουμε :

$$e_i(X) = q(X)\, g(X) + p(X)$$

όπου p(X) είναι το υπόλοιπο της διαίρεσης βαθμού n – k – 1 ή μικρότερου. Προσθέτοντας το $e_p(X)$ και στα δύο μέλη της παραπάνω σχέσης έχουμε

$$e(X) = e_i(X) + e_p(X) = q(X)\, g(X) + p(X) + e_p(X)$$

Από τη στιγμή που το $e_p(X)$ έχει βαθμό n – k – 1 ή μικρότερο, το $p(X) + e_p(X)$ θα πρέπει να είναι το υπόλοιπο που προκύπτει από τη διαίρεση του e (X) με το γενεσιουργό (generator) πολυώνυμο. Επομένως το

$p(X) + e_p(X)$ θα ισούται με το σύνδρομο του λαμβανόμενου διανύσματος r(X)

$$s(X) = p(X) + e_p(X)$$

Από την παραπάνω σχέση έχουμε :

$$e_p(X) = s(X) + p(X)$$

Αυτό σημαίνει ότι αν το pattern $e_i(X)$ που περιλαμβάνει τα λάθη στις θέσεις του μηνύματος είναι γνωστό τότε μπορούν να υπολογιστούν τα λάθη στις θέσεις ισοτιμίας, δηλαδή το pattern λάθους $e_p(X)$.

Η αποκωδικοποίηση Kasami για την παγίδευση λάθους απαιτεί την εύρεση ενός σετ πολυωνύμων $[\varphi_j(X)]_{j=1}^{N}$ βαθμού κ – 1 ή μικρότερου τέτοια ώστε για κάθε διορθώσιμο pattern λάθους e(X), υπάρχει ένα πολυώνυμο $\varphi_j(X)$ για το οποίο το $X^{n-k}\,\varphi_j(X)$ αντιστοιχεί στο τμήμα του μηνύματος του e(X) ή στο τμήμα του μηνύματος μιας κυκλικής μετακίνησης του e(X).

Τα πολυώνυμα $\varphi_j(X)$ ονομάζονται συνοδευτικά πολυώνυμα(covering polynomials). Έστω $p_j(X)$ είναι το υπόλοιπο της διαίρεσης του $X^{n-k}\,\varphi_j(X)$ με το γενεσιουργό (generator) πολυώνυμο g(X) του κώδικα.

Η διαδικασία αποκωδικοποίησης περιγράφεται με τα επόμενα βήματα :

Βήμα 1 Υπολογίζεται το σύνδρομο s(X) με το να εισάγουμε ολόκληρο το λαμβανόμενο διάνυσμα μέσα στο καταχωρητή συνδρόμου.

Βήμα 2 Υπολογίζεται το βάρος του αθροίσματος $s(X) + p_j(X)$ για κάθε j = 1,2,…N(δηλαδή το $w[s(X) + p_j(X)]$)

Βήμα 3 Αν για κάποιο l το $w[s(X) + p_l(X)] \leq t - w[\varphi_l(X)]$ τότε το $X^{n-k}\varphi_l(X)$ είναι το pattern λάθους του τμήματος του μηνύματος του e(X) και το $s(X) + p_l(X)$ είναι το pattern λάθους του τμήματος του ισοτιμίας του e(X). Επομένως

$e(X) = s(X) + p_l(X) + X^{n-k}\varphi_l(X)$.

Η διόρθωση επιτυγχάνεται με το modulo-2 άθροισμα r(X) + e(X). Το βήμα αυτό απαιτεί N(n –k) εισόδους- πύλες κατωφλίου για να ελέγξει τα βάρη του $s(X) + p_j(X)$ για j = 1,2,…N.

Βήμα 4 Αν $w[s(X) + p_j(X)] > t - w[\varphi_j(X)]$ για όλα τα j = 1,2,…N οι καταχωρητές συνδρόμου και buffer μετακινούνται κυκλικά μια φορά. Τότε τα περιεχόμενα του καταχωρητή συνδρόμου $s^{(1)}(X)$ είναι το σύνδρομο που αντιστοιχεί στο $e^{(1)}(X)$ που είναι το pattern λάθους που παίρνουμε με μια κυκλική μετακίνηση του e(X) προς τα δεξιά.

Βήμα 5 Το βάρος του $s^{(1)}(X) + p_j(X)$ υπολογίζεται για j = 1,2,…N. Αν για κάποιο l το

$w[s^{(1)}(X) + p_l(X)] \leq t - w[\varphi_l(X)]$ τότε το $X^{n-k}\varphi_l(X)$ περιέχει τα λάθη του τμήματος του μηνύματος του $e^{(1)}(X)$ και το $s^{(1)}(X) + p_l(X)$ περιέχει τα λάθη του τμήματος του ισοτιμίας του

$e^{(1)}(X)$. Επομένως $e(X) = s^{(1)}(X) + p_l(X) + X^{n-k}\varphi_l(X)$.

Η διόρθωση επιτυγχάνεται με το modulo-2 άθροισμα $r^{(1)}(X) + e^{(1)}(X)$. Αν $w[s^{(1)}(X) + p_j(X)] > t - w[\varphi_j(X)]$ για όλα τα j = 1,2,…N οι καταχωρητές συνδρόμου και buffer μετακινούνται κυκλικά άλλη μια φορά.

Βήμα 6 Οι καταχωρητές συνδρόμου και buffer συνεχώς μετακινούνται ως ότου βρεθεί το $s^{(i)}(X)$ για κάποιο l για το οποίο $A = w[s^{(i)}(X) + p_l(X)] \leq t - w[\varphi_l(X)]$. Τότε το

$e^{(i)}(X) = s^{(i)}(X) + p_l(X) + X^{n-k}\varphi_l(X)$.

Αν η ποσότητα A δεν κατέβει ποτέ κάτω από t - w[$\varphi_j(X)$] για κάθε j τη στιγμή που οι καταχωρητές συνδρόμου και buffer έχουν κυκλικά μετακινηθεί n – 1 φορές τότε ένα μη διορθώσιμο pattern λάθους έχει προκύψει.

5.3 Κώδικας Golay

Ο (23, 12) κώδικας Golay μπορεί να διορθώσει ένα οποιοδήποτε συνδυασμό τριών ή λιγότερων τυχαίων λαθών σε ένα μπλοκ 23 ψηφίων. Παράγεται είτε από το $g_1(X)$ είτε από το $g_2(X)$ τα οποία είναι :

$g_1(X) = 1 + X^2 + X^4 + X^5 + X^6 + X^{10} + X^{11}$

$g_2(X) = 1 + X + X^5 + X^6 + X^7 + X^9 + X^{11}$

Τα $g_1(X)$, $g_2(X)$ είναι και τα δύο παράγοντες του $X^{23} + 1$ και $X^{23} + 1 = (1+X)g_1(X)g_2(X)$. Η αποκωδικοποίηση γίνεται από ένα 11 – βαθμίδων καταχωρητή ολίσθησης με συνδέσεις ανάδρασης είτε στο $g_1(X)$ είτε στο $g_2(X)$. Για να αποκωδικοποιήσουμε τον κώδικας Golay(23, 12) με t =3 η κυριότερη μέθοδος βασίζεται στον Kasami αποκωδικοποιητή.

Το σετ των πολυωνύμων $[\varphi_j(X)]_{j=1}^{N}$ επιλέγεται ως ακολούθως :

$\varphi_1(X) = 0$, $\varphi_2(X) = X^5$, $\varphi_3(X) = X^6$.

Έστω $g_1(X) = 1 + X^2 + X^4 + X^5 + X^6 + X^{10} + X^{11}$ είναι το γενεσιουργό (generator) πολυώνυμο. Διαιρώντας το $X^{11}\varphi_j(X)$

με το $g_1(X)$ για j = 1,2,…N παίρνουμε τα ακόλουθα υπόλοιπα :

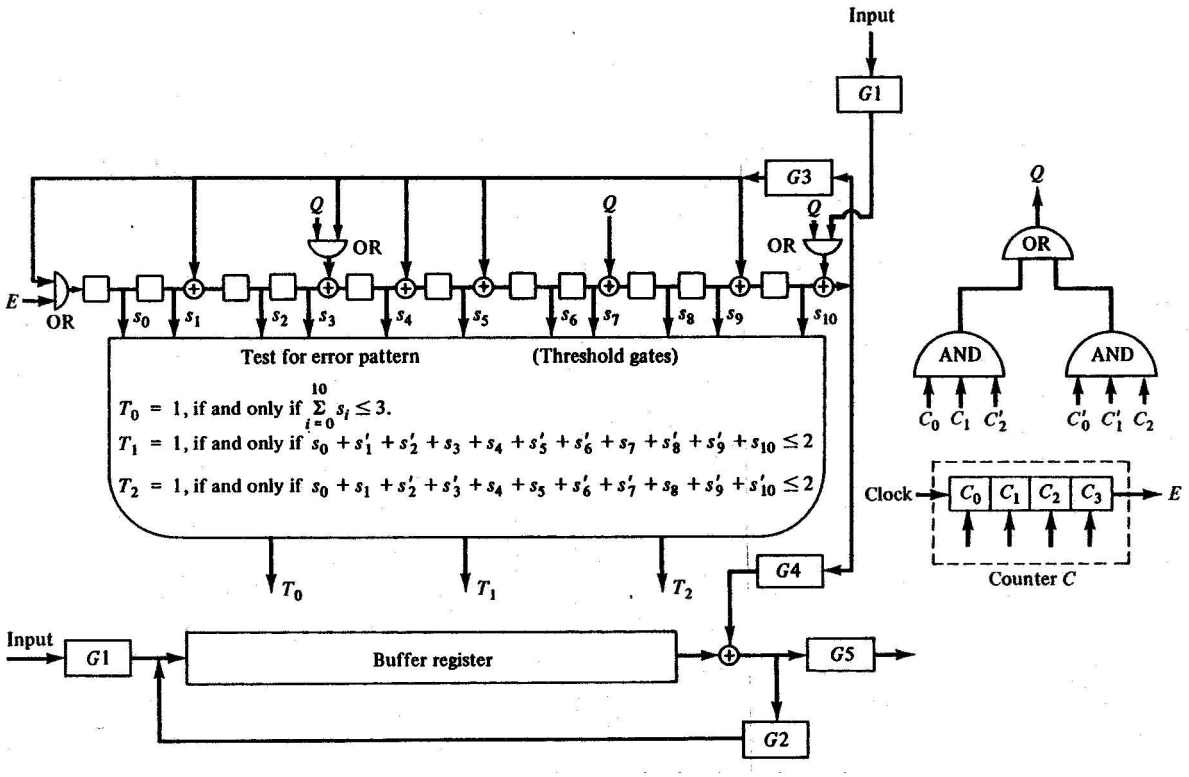
$\rho_1(X) = 0$,

$\rho_2(X) = X + X^2 + X^5 + X^6 + X^8 + X^9$

$\rho_3(X) = X\,\rho_2(x) = X^2 + X^3 + X^6 + X^7 + X^9 + X^{10}$

Ένας αποκωδικοποιητής βασισμένος στην μέθοδο Kasami παγίδευσης λάθους φαίνεται στο σχήμα

5.4.



Σχήμα 5.5 Αποκωδικοποιητής παγίδευσης λαθών για τον Golay κώδικα

Το λαμβανόμενο διάνυσμα r(X) = r_0+r_1X +r_2X^2 +…+r_{22}X22 μεταφέρεται στον καταχωρητή συνδρόμου από το δεξιό άκρο το οποίο είναι ισοδύναμο με μια αρχική κυκλική ολίσθηση του λαμβανόμενου διανύσματος 11 φορές. Τα περιεχόμενα του καταχωρητή συνδρόμου τότε θα αντιστοιχούν στο $r^{(11)}(X)$. Σε αυτήν την περίπτωση αν τα λάθη είναι περιορισμένα στις πρώτες 11 υψηλής τάξης θέσεις X^{12}, X^{13}, ..., X^{22} του r(X) το σύνδρομο αντιστοιχεί στα λάθη αυτών των θέσεων. Η διαδικασία διόρθωσης λαθών αυτού του αποκωδικοποιητή περιγράφεται από τα ακόλουθα βήματα :

Βήμα 1 Οι πύλες 1,3,5 είναι ανοικτές και οι πύλες 2,4 κλειστές. Το λαμβανόμενο διάνυσμα r(X) διαβάζεται ταυτόχρονα στον καταχωρητή συνδρόμου και στον buffer. Το

s(X) = s_0+s_1X+…+$s_{10}X^{10}$ διαμορφώνεται και διαβάζεται στις 3 πύλες κατωφλίου.

Βήμα 2 Οι πύλες 1,4,5 κλείνουν και οι 2,3 ανοίγουν. Το σύνδρομο ελέγχεται για διορθώσιμα patterns λάθους ως ακολούθως :

α) Αν το $w[s(X)] \leq 3$, όλα τα λάθη βρίσκονται μαζεμένα στις 11 υψηλής τάξης θέσεις του r(X) και το s(X) αντιστοιχεί στα λάθη. Επομένως τα λανθασμένα σύμβολα είναι τα επόμενα 11 ψηφία που θα βγουν από τον καταχωρητή buffer. Η έξοδος της πύλης κατωφλίου T_o ενεργοποιεί την πύλη 4 και απενεργοποιεί την πύλη 3. Τα ψηφία διαβάζονται ένα κάθε φορά που βγαίνουν από τον καταχωρητή buffer. Το κάθε ψηφίο που βγαίνει από τον καταχωρητή συνδρόμου προστίθεται (modulo 2) με το ψηφίο που βγαίνει από τον καταχωρητή buffer και έτσι διορθώνονται τα λάθη.

β) Αν το $w[s(X)] > 3$ το βάρος του $s(X) + p_2(X)$ ελέγχεται. Αν $w[s(X) + p_2(X)] \leq 2$ τότε

$$s(X) + p_2(X) = s_0 + s'_1X + s'_2X^2 + s_3X^3 + s_4X^4 + s'_5X^5 + s'_6X^6 + s_7X^7 + s'_8X^8 + s'_9X^9 + s_{10}X^{10}$$

αντιστοιχεί στο pattern λάθους των 11 υψηλής τάξης θέσεων της λαμβανόμενης λέξης και ένα απλό λάθος συμβαίνει στη θέση X^5, όπου το s'_i είναι το συμπληρωματικό του s_i. Η πύλη 4 ενεργοποιείται και η πύλη 3 απενεργοποιείται. Ο μετρητής αρχίζει τη μέτρηση από το 2. Την ίδια στιγμή ο καταχωρητής συνδρόμου ολισθαίνει χωρίς ανάδραση. Η έξοδος Q που είναι 1 όταν και μόνο όταν ο μετρητής μετρά 3 και 4, τροφοδοτεί τον καταχωρητή συνδρόμου για να διαμορφώσει το pattern λάθους $s(X) + p_2(X)$. Όταν ο μετρητής φτάσει στο 8 η έξοδος E είναι 1 και η πιο αριστερή βαθμίδα του καταχωρητή συνδρόμου ορίζεται 1. Αυτό το 1 χρησιμοποιείται για την διόρθωση του λάθους στη θέση X^5 του λαμβανόμενου διανύσματος r(X). Τα ψηφία που εξέρχονται από τον buffer διορθώνονται από τα ψηφία που εξέρχονται από τον καταχωρητή συνδρόμου.

γ) Αν το $w[s(X)] > 3$ και $w[s(X) + p_2(X)] > 2$, το βάρος του $s(X) + p_3(X)$ ελέγχεται. Αν το $w[s(X) + p_3(X)] \leq 2$ τότε $s(X) + p_3(X) = s_0 + s_1X + s'_2X^2 + s'_3X^3 + s_4X^4 + s_5X^5 + s'_6X^6 + s'_7X^7 + s_8X^8 + s'_9X^9 + s'_{10}X^{10}$ αντιστοιχεί στο pattern λάθους των 11 υψηλής τάξης θέσεων της λαμβανόμενης λέξης και ένα απλό λάθος συμβαίνει στη θέση X^6.Η διόρθωση είναι η ίδια με αυτή του βήματος (β) εκτός του ότι ο μετρητής ξεκινά τη μέτρηση από το 3.Αν το $w[s(X)] > 3$ και το $w[s(X) + p_2(X)] > 2$ και το $w[s(X) + p_3(X)] > 2$, ο αποκωδικοποιητής προχωρά στο βήμα 3.

Βήμα 3 Οι καταχωρητές συνδρόμου και buffer ολισθαίνουν μία φορά με τις πύλες 1,4,5 απενεργοποιημένες και τις πύλες 2 και 3 ενεργοποιημένες. Τα νέα περιεχόμενα του καταχωρητή συνδρόμου είναι το $s^{(1)}(X)$. Το βήμα 2 τότε επαναλαμβάνεται.

Βήμα 4 η διαδικασία αποκωδικοποίησης ολοκληρώνεται μόλις ο καταχωρητής συνδρόμου έχει κυκλικά ολισθήσει 46 φορές. Πύλη 5 τότε ενεργοποιείται και το διάνυσμα στον buffer μεταφέρεται έξω στα δεδομένα.

Κεφάλαιο 6 – BCH Κώδικες

Οι BCH Κώδικες (Bose ,Chaudhuri, Hocquenghem) διαμορφώνουν μια πολύ αξιόλογη κλάση κωδίκων για διόρθωση τυχαίων λαθών. Αυτή η κλάση είναι μια αξιόλογη γενίκευση των κωδίκων Hamming για τη διόρθωση πολλαπλών λαθών. Στο κεφάλαιο αυτό παρουσιάζεται μια πολύ σημαντική υποκλάση των δυαδικών BCH Κωδίκων τόσο ως προς τη θεωρία όσο και ως προς την υλοποίηση τους.

6.1 Περιγραφή των BCH Κωδίκων

Για οποιουσδήποτε θετικούς ακέραιους m ($m \geq 3$) και t ($t < 2^m - 1$) υπάρχει ένας δυαδικός BCH κώδικας με τις ακόλουθες παραμέτρους :

Μήκος μπλοκ : $n = 2^m - 1$

Αριθμός των ψηφίων ελέγχου ισοτιμίας : $n - k \leq m \cdot t$

Ελάχιστη απόσταση : $dmin \geq 2t + 1$

Ο κώδικας αυτός μπορεί να διορθώσει κάθε συνδυασμό από t ή λιγότερα λάθη σε ένα μπλοκ από

$n = 2^m - 1$ ψηφία και ονομάζεται t – error – correcting BCH κώδικας. Το γενεσιουργό (generator) πολυώνυμο αυτού του κώδικα καθορίζεται από όρους που είναι οι ρίζες από το Galois πεδίο $GF(2^m)$.

Έστω a είναι ένα βασικό (**primitive**) στοιχείο του $GF(2^m)$. Το γενεσιουργό (generator) πολυώνυμο g(X) ενός t – error – correcting BCH κώδικα μήκους $2^m - 1$ είναι το μικρότερου βαθμού πολυώνυμο ορισμένο στο GF(2) το οποίο έχει ρίζες τα $a, a^2, a^3, ..., a^{2t}$ (6.1)

(δηλ. το $g(a^i) = 0$ για $1 \leq i \leq 2t$) καθώς και τις συζυγείς τους. Έστω $\varphi_i(X)$ είναι το ελάχιστο πολυώνυμο(minimal polynomial) του a^i.Τότε το g(X) πρέπει να είναι το ελάχιστο κοινό πολλαπλάσιο των $\varphi_1(X), \varphi_2(X), \varphi_3(X),..., \varphi_{2t}(X)$ δηλαδή

$g(X) = LCM \{ \varphi_1(X), \varphi_2(X), \varphi_3(X),..., \varphi_{2t}(X) \}$ (6.2)

Αν i είναι ένας ζυγός ακέραιος μπορεί να εκφραστεί ως γινόμενο της ακόλουθης μορφής :

$i = i' 2^r$ όπου i' είναι ένας περιττός ακέραιος και $r > 1$. Τότε $a^i = (a^{i'})^{2^r}$ είναι η συζυγής της $a^{i'}$ και επομένως η a^i και $a^{i'}$ θα έχουν το ίδιο ελάχιστο πολυώνυμο δηλαδή

$\varphi_i(X) = \varphi_{i'}(X)$

Αφού κάθε άρτια δύναμη του a της σειράς (6.1) έχει το ίδιο ελάχιστο πολυώνυμο με κάποια προηγούμενη μονή δύναμη του a της σειράς ,

Άρα το g(X) μπορεί να ελαττωθεί :

$g(X) = \text{LCM} \{ \varphi_1(X), \varphi_3(X), ..., \varphi_{2t-1}(X) \}$ (6.3)

Αφού ο βαθμός του κάθε ελάχιστου πολυωνύμου είναι μικρότερος ή ίσος με m τότε και ο βαθμός του g(X) είναι το πολύ $m \cdot t$. Άρα και ο βαθμός των ψηφίων ελέγχου ισοτιμίας , $n - k$, του κώδικα είναι το πολύ $m \cdot t$. Δεν υπάρχει απλή φόρμα για την απαρίθμηση του $n - k$ αλλά αν το t είναι μικρό τότε το $n - k$ είναι ακριβώς ίσο με το $m \cdot t$. Οι παράμετροι για όλους τους δυαδικούς BCH κώδικες μήκους $2^m - 1$ με $m \leq 10$ δίνονται στον παρακάτω πίνακα

Πίνακας 6.1 BCH κώδικες που δημιουργούνται από τα primitive στοιχεία τάξης μικρότερης από 2^{10}

n	k	t	n	k	t	n	k	t
7	4	1	255	163	12	511	268	29
15	11	1		155	13		259	30
	7	2		147	14		250	31
	5	3		139	15		241	36
31	26	1		131	18		238	37
	21	2		123	19		229	38
	16	3		115	21		220	39
	11	5		107	22		211	41
	6	7		99	23		202	42
63	57	1		91	25		193	43

	51	2		87	26		184	45
	45	3		79	27		175	46
	39	4		71	29		166	47
	36	5		63	30		157	51
	30	6		55	31		148	53
	24	7		47	42		139	54
	18	10		45	43		130	55
	16	11		37	45		121	58
	10	13		29	47		112	59
7	15		21	55		103	61	
127	120	1		13	59		94	62
	113	2		9	63		85	63
	106	3	511	502	1		76	85
	99	4		493	2		67	87
	92	5		484	3		58	91
	85	6		475	4		49	93
	78	7		466	5		40	95
	71	9		457	6		31	109
	64	10		448	7		28	111
	57	11		439	8		19	119
	50	13		430	9		10	121
	43	14		421	10	1023	1013	1
	36	15		412	11		1003	2
	29	21		403	12		993	3
	22	23		394	13		983	4
	15	27		385	14		973	5
	8	31		376	15		963	6
255	247	1		367	16		953	7
	239	2		358	18		943	8
	231	3		349	19		933	9
	223	4		340	20		923	10
	215	5		331	21		913	11
	207	6		322	22		903	12

	199	7		313	23		893	13
	191	8		304	25		883	14
	187	9		295	26		873	15
	179	10		286	27		863	16
	171	11		277	28		858	17
1023	848	18	1023	553	52	1023	268	103
	838	19		543	53		258	106
	828	20		533	54		248	107
	818	21		523	55		238	109
	808	22		513	57		228	110
	798	23		503	58		218	111
	788	24		493	59		208	115
	778	25		483	60		203	117
	768	26		473	61		193	118
	758	27		463	62		183	119
	748	28		453	63		173	122
	738	29		443	73		163	123
	728	30		433	74		153	125
	718	31		423	75		143	126
	708	34		413	77		133	127
	698	35		403	78		123	170
	688	36		393	79		121	171
	678	37		383	82		111	173
	668	38		378	83		101	175
	658	39		368	85		91	181
	648	41		358	86		86	183
	638	42		348	87		76	187
	628	43		338	89		66	189
	618	44		328	90		56	191
	608	45		318	91		46	219
	598	46		308	93		36	223
	588	47		298	94		26	239
	578	49		288	95		16	147

573	50		278	102		11	255
563	51						

Αυτοί οι κώδικες καλούνται βασικοί (primitive) BCH κώδικες.

Από τη σχέση (6.3) βλέπουμε ότι για ένα BCH κώδικα μήκους $2^m - 1$ που διορθώνει ένα απλό λάθος το πολυώνυμο g(X) έχει τη μορφή $g(X) = \varphi_1(X)$. Από τη στιγμή που το α είναι ένα βασικό (primitive) στοιχείο του $GF(2^m)$, $\varphi_1(X)$ είναι ένα βασικό (primitive) πολυώνυμο βαθμού m. Επομένως ο BCH κώδικας μήκους $2^m - 1$ για τη διόρθωση ενός απλού λάθους είναι ένας Hamming κώδικας.

Χρησιμοποιώντας το βασικό (primitive) πολυώνυμο (πρόγονος) $\rho(X) = 1+X+X^6$ μπορούμε να δημιουργήσουμε το πεδίο Galois $GF(2^6)$ που φαίνεται στο πίνακα 6.2.Τα ελάχιστα πολυώνυμα των στοιχείων του $GF(2^6)$ απαριθμούνται στον πίνακα 6.3.Χρησιμοποιώντας τη σχέση (6.3) βρίσκουμε όλα τα γενεσιουργά πολυώνυμα όλων των BCH κωδίκων μήκους 63 τα οποία φαίνονται στον πίνακα 6.4.

Πίνακας 6.2 Πεδίο Galois $GF(2^6)$ με $\rho(a) = 1+a+a^6 = 0$

0	0	(0 0 0 0 0 0)
1	1	(1 0 0 0 0 0)
a	a	(0 1 0 0 0 0)
a^2	a^2	(0 0 1 0 0 0)
a^3	a^3	(0 0 0 1 0 0)
a^4	a^4	(0 0 0 0 1 0)
a^5	a^5	(0 0 0 0 0 1)
a^6	$1 + a$	(1 1 0 0 0 0)
a^7	$a + a^2$	(0 1 1 0 0 0)
a^8	$a^2 + a^3$	(0 0 1 1 0 0)
a^9	$a^3 + a^4$	(0 0 0 1 1 0)
a^{10}	$a^4 + a^5$	(0 0 0 0 1 1)

a^{11} $1+a+a^5$ (1 1 0 0 0 1)

a^{12} $1+a^2$ (1 0 1 0 0 0)

a^{13} $a+a^3$ (0 1 0 1 0 0)

a^{14} a^2+a^4 (0 0 1 0 1 0)

a^{15} a^3+a^5 (0 0 0 1 0 1)

a^{16} $1+a+a^4$ (1 1 0 0 1 0)

a^{17} $a+a^2+a^5$ (0 1 1 0 0 1)

a^{18} $1+a+a^2+a^3$ (1 1 1 1 0 0)

a^{19} $a+a^2+a^3+a^4$ (0 1 1 1 1 0)

a^{20} $a^2+a^3+a^4+a^5$ (0 0 1 1 1 1)

a^{21} $1+a+a^3+a^4+a^5$ (1 1 0 1 1 1)

a^{22} $1+a^2+a^4+a^5$ (1 0 1 0 1 1)

a^{23} $1+a^3+a^4+a^5$ (1 0 0 1 0 1)

a^{24} $1+a^4$ (1 0 0 0 1 0)

a^{25} $a+a^5$ (0 1 0 0 0 1)

a^{26} $1+a+a^2$ (1 1 1 0 0 0)

a^{27} $a+a^2+a^3$ (0 1 1 1 0 0)

a^{28} $a^2+a^3+a^4$ (0 0 1 1 1 0)

a^{29} $a^3+a^4+a^5$ (0 0 0 1 1 1)

a^{30} $1+a+a^4+a^5$ (1 1 0 0 1 1)

a^{31} $1+a^2+a^5$ (1 0 1 0 0 1)

a^{32} $1+a^3$ (1 0 0 1 0 0)

a^{33} $a+a^4$ (0 1 0 0 1 0)

a^{34} a^2+a^5 (0 0 1 0 0 1)

a^{35} $1+a+a^3$ (1 1 0 1 0 0)

a^{36} $a+a^2+a^4$ (0 1 1 0 1 0)

a^{37} $a^2+a^3+a^5$ (0 0 1 1 0 1)

a^{38} $1+a+a^3+a^4$ (1 1 0 1 1 0)

a^{39} $a+a^2+a^4+a^5$ (0 1 1 0 1 1)

a^{40} $1+a+a^2+a^3+a^5$ (1 1 1 1 0 1)

a^{41} $1+a^2+a^3+a^4$ (1 0 1 1 1 0)

a^{42} $a+a^3+a^4+a^5$ (0 1 0 1 1 1)

a^{43} $1+a+a^2+a^4+a^5$ (1 1 1 0 1 1)

a^{44}	1+	a^2 +	a^3 +		a^5	(1 0 1 1 0 1)
a^{45}	1+	+	a^3 +	a^4		(1 0 0 1 1 0)
a^{46}	a+		+	a^4 +	a^5	(0 1 0 0 1 1)
a^{47}	1+a+	a^2 +			a^5	(1 1 1 0 0 1)
a^{48}	1+	a^2 +	a^3			(1 0 1 1 0 0)
a^{49}	a+		a^3 +	a^4		(0 1 0 1 1 0)
a^{50}		a^2 +		a^4 +	a^5	(0 0 1 0 1 1)
a^{51}	1+a+		a^3 +		a^5	(1 1 0 1 0 1)
a^{52}	1+	a^2 +		a^4		(1 0 1 0 1 0)
a^{53}	a+		a^3 +		a^5	(0 1 0 1 0 1)
a^{54}	1+a+	a^2 +		a^4		(1 1 1 0 1 0)
a^{55}	a+	a^2 +	a^3 +		a^5	(0 1 1 1 0 1)
a^{56}	1+a+	a^2 +	a^3 +	a^4		(1 1 1 1 1 0)
a^{57}	a+	a^2 +	a^3 +	a^4 +	a^5	(0 1 1 1 1 1)
a^{58}	1+ a+	a^2 +	a^3 +	a^4 +	a^5	(1 1 1 1 1 1)
a^{59}	1+	a^2 +	a^3 +	a^4 +	a^5	(1 0 1 1 1 1)
a^{60}	1+		a^3 +	a^4 +	a^5	(1 0 0 1 1 1)
a^{61}	1+			a^4 +	a^5	(1 0 0 0 1 1)
a^{62}	1+				a^5	(1 0 0 0 0 1)

$a^{63} = 1$

Πίνακας 6.3 Ελάχιστα πολυώνυμα των στοιχείων του GF(2^6)

Στοιχεία	Ελάχιστα πολυώνυμα
$\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^{32}$	$1+X+X^6$
$\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{48}, \alpha^{33}$	$1+X+X^2+X^4+X^6$
$\alpha^5, \alpha^{10}, \alpha^{20}, \alpha^{40}, \alpha^{17}, \alpha^{34}$	$1+X+X^2+X^5+X^6$

$\alpha^{7}, \alpha^{14}, \alpha^{28}, \alpha^{56}, \alpha^{49}, \alpha^{35}$	$1+X^3+X^6$
$\alpha^{9}, \alpha^{18}, \alpha^{36}$	$1+X^2+X^3$
$\alpha^{11}, \alpha^{22}, \alpha^{44}, \alpha^{25}, \alpha^{50}, \alpha^{37}$	$1+X^2+X^3+X^5+X^6$
$\alpha^{13}, \alpha^{26}, \alpha^{52}, \alpha^{41}, \alpha^{19}, \alpha^{38}$	$1+X+X^3+X^4+X^6$
$\alpha^{15}, \alpha^{30}, \alpha^{60}, \alpha^{57}, \alpha^{51}, \alpha^{39}$	$1+X^2+X^4+X^5+X^6$
α^{21}, α^{42}	$1+X+X^2$
$\alpha^{23}, \alpha^{46}, \alpha^{29}, \alpha^{58}, \alpha^{53}, \alpha^{43}$	$1+X+X^4+X^5+X^6$
$\alpha^{27}, \alpha^{54}, \alpha^{45}$	$1+X+X^3$
$\alpha^{31}, \alpha^{62}, \alpha^{61}, \alpha^{59}, \alpha^{55}, \alpha^{47}$	$1+X^5+X^6$

Πίνακας 6.4 Γεννεσιουργά πολυώνυμα όλων των BCH κωδίκων μήκους 63

n	k	t	g(X)
63	57	1	$g_1(X)=1+X+X^6$
	51	2	$g_2(X)=(1+X+X^6)(1+X+X^2+X^4+X^6)$
	45	3	$g_3(X)= (1+X+X^2+X^5+X^6) \cdot g_2(X)$
	39	4	$g_4(X)= (1 +X^3+ X^6) \cdot g_3(X)$
	36	5	$g_5(X)= (1 +X^2+X^3) \cdot g_4(X)$
	30	6	$g_6(X)= (1+X^2+X^3+X^5+X^6) \cdot g_5(X)$
	24	7	$g_7(X)= (1+X+X^3+X^4+X^6) \cdot g_6(X)$
	18	10	$g_{10}(X)= (1+X^2+X^4+X^5+X^6) \cdot g_7(X)$
	16	11	$g_{11}(X)= (1+X +X^2) \cdot g_{10}(X)$
	10	13	$g_{13}(X)= (1+X+ X^4+X^5+X^6) \cdot g_{11}(X)$
	7	15	$g_{15}(X)= (1+X +X^3) \cdot g_{13}(X)$

Έστω ότι $v(X) = v_o + v_1X + \ldots + v_{n-1}X^{n-1}$ είναι ένα πολυώνυμο με συντελεστές από το GF(2).

Αν το v(X) έχει ρίζες τα a, a^2,a^3, ...,a^{2t} τότε το v(X) θα διαιρείται από τα ελάχιστα πολυώνυμα $\varphi_1(X)$, $\varphi_2(X)$, $\varphi_3(X)$,..., $\varphi_{2t}(X)$ των a, a^2,a^3, ...,a^{2t} και θα διαιρείται και από το ελάχιστο κοινό πολλαπλάσιο τους (το γεννεσιουργό πολυώνυμο)

$g(X) = LCM \{ \varphi_1(X), \varphi_2(X), \varphi_3(X),..., \varphi_{2t}(X) \}$.

Επομένως το v(X) είναι ένα κωδικό πολυώνυμο. Άρα μπορούμε να ορίσουμε ένα t – error – correcting BCH κώδικα μήκους $2^m - 1$ με τον ακόλουθο τρόπο :

Ένα δυαδικό n – tuple $v = (v_o , v_1, v_2,\ldots , v_{n-1})$ είναι μία κωδική λέξη αν και μόνο αν το πολυώνυμο $v(X) = v_o + v_1X + \ldots + v_{n-1}X^{n-1}$ έχει τα a, a^2,a^3, ...,a^{2t} ρίζες. Αυτός ο ορισμός είναι πολύ χρήσιμος όσον αφορά την απόδειξη της ελάχιστης απόστασης αυτού του κώδικα.

Έστω $v(X) = v_o + v_1X + \ldots + v_{n-1}X^{n-1}$ είναι ένα κωδικό πολυώνυμο ενός BCH κώδικα μήκους $2^m - 1$ που διορθώνει t λάθη. Από τη στιγμή που a^i είναι ρίζα του v(X) για $1 \leq i \leq 2t$ τότε

$$v(a^i) = v_o + v_1 a^i + \ldots + v_{n-1} a^{i(n-1)} = 0 \quad (6.4)$$

Η παραπάνω ισότητα μπορεί να γραφτεί και με τη μορφή ενός γινομένου πινάκων ως ακολούθως :

$$(v_0, v_1, \ldots, v_{n-1}) \cdot \begin{bmatrix} 1 \\ a^i \\ a^{2i} \\ . \\ . \\ . \\ a^{(n-1)i} \end{bmatrix} = 0 \quad (6.5)$$

Για 1<= i <=2t

Η συνθήκη που δίνεται στη σχέση (6.5) απλώς αναφέρει ότι το εσωτερικό γινόμενο του (v_o , v_1, v_2,… , v_{n-1}) και (1, a^i ,…, $a^{i(n-1)}$) ισούται με το μηδέν. Το επόμενο βήμα είναι η διαμόρφωση του παρακάτω πίνακα :

$$H = \begin{bmatrix} 1 & a & a^2 & a^3 & . & . & . & a^{n-1} \\ 1 & a^2 & a^4 & a^6 & . & . & . & a^{2(n-1)} \\ 1 & a^3 & a^6 & a^9 & . & . & . & a^{3(n-1)} \\ . & & & & & & & . \quad . \\ . & & & & & & & . \\ . & & & & & & & . \\ 1 & a^{2t} & a^{4t} & a^{6t} & . & . & . & a^{2t(n-1)} \end{bmatrix} \quad (6.6)$$

προκύπτει από την (6.5) ότι αν v = (v_o , v_1, v_2,… , v_{n-1}) είναι κωδική λέξη σε ένα t – error – correcting BCH τότε :

$$\mathbf{v \cdot H^T = 0} \quad \mathbf{(6.7)}$$

Αντιστρόφως αν ένα n – tuple v = (v_o , v_1, v_2,… , v_{n-1}) ικανοποιεί τη συνθήκη (6.7) προκύπτει από τις (6.5) και (6.4)ότι για 1<= i <=2t , a^i είναι ρίζα του πολυωνύμου v(X). Επομένως το v πρέπει να είναι μια κωδική λέξη σε ένα t – error – correcting BCH κώδικα. Άρα ο κώδικας είναι ο δυαδικός χώρος του πίνακα **H** και ο **H** είναι ο parity – check πίνακας του κώδικα. Αν για κάποια i, j το α^j είναι συζυγής του α^i ,τότε το $v(\alpha^j)$ =0 αν και μόνο αν το $v(\alpha^i)$ =0.Από το παραπάνω προκύπτει ότι το εσωτερικό γινόμενο του v = (v_o , v_1, v_2,… , v_{n-1}) με τη i γραμμή του **H** είναι μηδέν και ότι το εσωτερικό γινόμενο του v με τη j γραμμή του **H** είναι επίσης μηδέν . Για αυτό το

λόγο η j γραμμή του **H** μπορεί να παραληφθεί . Ως αποτέλεσμα ο **H** πίνακας που δίνεται από τη σχέση (6.6) μπορεί να μειωθεί και να πάρει την ακόλουθη μορφή:

$$H = \begin{bmatrix} 1 & a & a^2 & a^3 & . & . & . & a^{n-1} \\ 1 & a^3 & a^6 & a^9 & . & . & . & a^{3(n-1)} \\ 1 & a^5 & a^{10} & a^{15} & . & . & . & a^{5(n-1)} \\ . & & & & & & & . \\ . & & & & & & & . \\ . & & & & & & & . \\ 1 & a^{2t-1} & a^{2(2t-1)} & a^{3(2t-1)} & . & . & . & a^{(2t-1)(n-1)} \end{bmatrix} \quad (6.8)$$

Παρατηρούμε ότι όλοι οι είσοδοι του **H** είναι στοιχεία του $GF(2^m)$. Κάθε στοιχείο στο $GF(2^m)$ μπορεί να αναπαρασταθεί με ένα m- tuple ορισμένο στο GF(2).Αν κάθε είσοδος του H αντικατασταθεί με το αντίστοιχο m- tuple ορισμένο στο GF(2) σε μορφή στήλης παίρνουμε το δυαδικό parity – check πίνακα του κώδικα.

Στο σημείο αυτό αποδεικνύεται ότι ο t – error – correcting BCH κώδικας που έχει οριστεί παραπάνω έχει ελάχιστη απόσταση τουλάχιστον 2t +1. Για να το αποδείξουμε θα πρέπει να δείξουμε ότι δεν υπάρχουν 2t ή λιγότερες στήλες του **H** που αν προστεθούν μας δίνουν άθροισμα μηδέν. Υποθέτουμε ότι υπάρχει ένα μη μηδενικό κωδικό διάνυσμα $v = (v_o , v_1, v_2, \dots , v_{n-1})$ με βάρος $\delta <= 2t$. Έστω $v_{j1}, v_{j2} ,..., v_{j\delta}$ είναι τα μη μηδενικά στοιχεία του $v(v_{j1}= v_{j2} = ...= v_{j\delta}=1)$. Βασιζόμενοι στη σχέση (6.6) και (6.7) θα έχουμε :

$0 = v \cdot H^T$

$$= (v_{j1}, v_{j2}, ..., v_{j\delta}) \cdot \begin{bmatrix} a^{j_1} & a^{2j_1} & . & . & . & a^{(2t)j_1} \\ a^{j_2} & a^{2j_2} & . & . & . & a^{(2t)j_2} \\ a^{j_3} & a^{2j_3} & . & . & . & a^{(2t)j_3} \\ . & . & & & & . \\ . & . & & & & . \\ . & . & & & & . \\ a^{j_\delta} & a^{2j_\delta} & . & . & . & \alpha^{(2t)j_\delta} \end{bmatrix}$$

$$= (1,1,..., \quad 1) \cdot \begin{bmatrix} a^{j_1} & a^{2j_1} & . & . & . & a^{(2t)j_1} \\ a^{j_2} & a^{2j_2} & . & . & . & a^{(2t)j_2} \\ a^{j_3} & a^{2j_3} & . & . & . & a^{(2t)j_3} \\ . & . & & & & . \\ . & . & & & & . \\ . & . & & & & . \\ a^{j_\delta} & a^{2j_\delta} & . & . & . & \alpha^{(2t)j_\delta} \end{bmatrix}$$

Η παραπάνω ισότητα οδηγεί στο συμπέρασμα ότι :

$$(1,1,..., \quad 1) \cdot \begin{bmatrix} a^{j_1} & a^{2j_1} & . & . & . & a^{(j_1)\delta} \\ a^{j_2} & a^{2j_2} & . & . & . & a^{(j_2)\delta} \\ a^{j_3} & a^{2j_3} & . & . & . & a^{(j_3)\delta} \\ . & . & & & & . \\ . & . & & & & . \\ . & . & & & & . \\ a^{j_\delta} & a^{2j_\delta} & . & . & . & \alpha^{(j_\delta)\delta} \end{bmatrix} = 0 \quad (6.9)$$

όπου ο δεύτερος πίνακας στα αριστερά είναι ένας $\delta \times \delta$ τετραγωνικός πίνακας . Για να ικανοποιείται η ισότητα (6.9) για $\delta=2t$ θα πρέπει η ορίζουσα του $\delta \times \delta$ να είναι μηδέν .

$$\begin{vmatrix} \alpha^{j_1} & a^{2j_1} & . & . & . & a^{\delta j_1} \\ a^{j_2} & a^{2j_2} & . & . & . & a^{\delta j_2} \\ a^{j_3} & a^{2j_3} & . & . & . & a^{\delta j_3} \\ . & & & & & . \\ . & & & & & . \\ . & & & & & . \\ a^{j_\delta} & a^{2j_\delta} & . & . & . & a^{\delta j_\delta} \end{vmatrix} = 0$$

Βγάζοντας τον κοινό παράγοντα από κάθε στήλη της ορίζουσας παίρνουμε :

$$a^{(j1+j2+\ldots+j\delta)} \cdot \begin{vmatrix} 1 & a^{j_1} & . & . & . & a^{(\delta-1)j_1} \\ 1 & a^{j_2} & . & . & . & a^{(\delta-1)j_2} \\ 1 & a^{j_3} & . & . & . & a^{(\delta-1)j_3} \\ . & . & & & & . \\ . & . & & & & . \\ . & . & & & & . \\ 1 & a^{j_\delta} & . & . & . & a^{(\delta-1)j_\delta} \end{vmatrix} = 0 \quad (6.10)$$

Η παραπάνω όμως ορίζουσα της ισότητας γνωστή και σαν Vandermonde ορίζουσα είναι μη μηδενική. Επομένως το γινόμενο που βρίσκεται αριστερά της ισότητας δεν μπορεί να γίνει μηδέν. Επομένως η αρχική υπόθεση στην οποία βασιστήκαμε ότι δηλαδή υπάρχει ένα μη μηδενικό κωδικό

διάνυσμα v βάρους $\delta \leq 2t$ δεν ισχύει. Συμπεραίνουμε λοιπόν ότι το ελάχιστο βάρος ενός t – error – correcting BCH κώδικα που ορίζεται παραπάνω είναι τουλάχιστον 2t+1.

Η παράμετρος 2t+1 ονομάζεται συνήθως σχεδιαστική απόσταση του t – error – correcting BCH κώδικα . Η πραγματική ελάχιστη απόσταση του t – error – correcting BCH κώδικα μπορεί όμως και να μην είναι 2t+1(σχεδιαστική απόσταση). Σε πολλές περιπτώσεις η πραγματική ελάχιστη απόσταση είναι μεγαλύτερη από την σχεδιαστική απόσταση.

Οι δυαδικοί BCH κώδικες με μήκος $n \neq 2^m - 1$ μπορούν να υλοποιηθούν με τον ίδιο τρόπο όπως στην περίπτωση $n = 2^m - 1$. Έστω β είναι ένα στοιχείο τάξης n στο πεδίο $GF(2^m)$. Γνωρίζουμε ότι το n είναι παράγοντας του $2^m - 1$. Έστω ότι g(X) είναι το δυαδικό πολυώνυμο ελάχιστου βαθμού που έχει ρίζες τα $\beta,\beta^2,...,\beta^{2t}$. Έστω $\psi_1(X)$, $\psi_2(X)$,..., $\psi_{2t}(X)$ τα ελάχιστα πολυώνυμα των $\beta,\beta^2,...,\beta^{2t}$ αντίστοιχα. Τότε το $g(X)= LCM \{ \psi_1(X) , \psi_2(X) ,..., \psi_{2t}(X) \}$.

Αφού $\beta^n = 1$, τα $\beta,\beta^2,...,\beta^{2t}$ είναι ρίζες του $X^n + 1$.Άρα και το g(X) θα είναι παράγοντας του $X^n + 1$. Ο κυκλικός κώδικας που παράγεται από το g(X) είναι ένας t – error –correcting BCH κώδικας μήκους n. Με παρόμοιο τρόπο μπορούμε να δείξουμε ότι τα parity – check ψηφία αυτού του κώδικα είναι το ανώτερο mt και η ελάχιστη απόσταση αυτού του κώδικα είναι τουλάχιστον 2t+1. Αν το β δεν είναι ένα βασικό (primitive) στοιχείο του $GF(2^m)$, $n \neq 2^m - 1$ και ο κώδικας ονομάζεται μη βασικός (non primitive) BCH κώδικας.

Στο σημείο αυτό δίνουμε ένα γενικό ορισμό των δυαδικών BCH κωδίκων. Έστω β είναι ένα στοιχείο του $GF(2^m)$ και l_0 ένας μη αρνητικός ακέραιος. Τότε ένας δυαδικός BCH κώδικας με σχεδιαστική απόσταση d_0 παράγεται από το δυαδικό πολυώνυμο g(X) ελάχιστου βαθμού που έχει ρίζες τις ακόλουθες συνεχείς δυνάμεις του β :

$\beta^{l_0},\beta^{l_0+1},...\beta^{l_0+d_0-2}$

Για $0 \leq i \leq d_o - 1$ έστω $\psi_i(X)$ το ελάχιστο πολυώνυμο τάξης β^{l_0+i}. Τότε το :

$g(X) = LCM \{ \psi_0(X) , \psi_1(X) , \psi_2(X) ,..., \psi_{do-2}(X) \}$

και το μήκος του κώδικα θα είναι :

$n = LCM\{n_0 ,n_1,\ldots,n_{do-2}\}$.

Ο BCH κώδικας που ορίστηκε παραπάνω έχει ελάχιστη απόσταση τουλάχιστον **do** και έχει όχι περισσότερα από m $(d_0 - 1)$ parity – check ψηφία. Ο κώδικας αυτός είναι σε θέση να διορθώνει

$[(d_0 - 1)/2]$ ή λιγότερα λάθη. Για τιμές $l_0 = 1$, **do** $= 2t+1$ και β βασικό στοιχείο του $GF(2^m)$, τότε ο κώδικας θα είναι ένας t – error –correcting βασικός (primitive) BCH κώδικας μήκους $2^m - 1$.

Για τιμές $l_0 = 1$, **do** $= 2t+1$ και β μη βασικό στοιχείο του $GF(2^m)$, τότε ο κώδικας θα είναι ένας μη βασικός (nonprimitive) t – error –correcting BCH κώδικας μήκους n που είναι η τάξη του β . Παρατηρούμε ότι στον ορισμό του BCH κώδικα με σχεδιαστική απόσταση d_0 , απαιτούμε ότι το γενεσιουργό (generator) πολυώνυμο g(X) θα πρέπει να έχει $(d_0 - 1)$ συνεχείς δυνάμεις ενός στοιχείου του πεδίου ως ρίζες και αυτή η απαίτηση εγγυάται ότι ο κώδικας έχει ελάχιστη απόσταση τουλάχιστον d_0.Το κατώτερο αυτό όριο της ελάχιστης απόστασης ονομάζεται BCH όριο.

6.2 Αποκωδικοποίηση των BCH κωδίκων

Υποθέτουμε ότι η κωδική λέξη όπου μεταδίδεται είναι η $v(X) = v_o + v_1X + \ldots + v_{n-1}X^{n-1}$ και τα λάθη μετάδοσης που συμβαίνουν μας δίνουν το ακόλουθο λαμβανόμενο διάνυσμα :

$$\mathbf{r}(X) = r_o + r_1X + \ldots + r_{n-1}X^{n-1} \quad .$$

Έστω e(X) είναι το error pattern. Τότε το

$$\mathbf{r}(X) = v(X) + e(X) \quad (6.11)$$

Ως συνήθως το πρώτο βήμα της αποκωδικοποίησης ενός κώδικα είναι ο υπολογισμός του συνδρόμου από το λαμβανόμενο διάνυσμα **r(**X). Για την αποκωδικοποίηση ενός t – error – correcting βασικού (primitive) BCH κώδικα, το σύνδρομο είναι ένα 2t- tuple ,

$$S = (S_1, S_2, \ldots, S_{2t}) = r \cdot H^T , \quad (6.12)$$

όπου H είναι ο πίνακας που δίνεται από την σχέση (6.6). Από την (6.6) και (6.12) βρίσκουμε ότι το ith στοιχείο του συνδρόμου είναι :

$$S_i = r(a^i) = r_o + r_1 a^i + \ldots + r_{n-1} a^{i(n-1)} \quad \text{για } 0 \le i \le 2t. \quad (6.13)$$

Παρατηρούμε ότι τα συστατικά του συνδρόμου είναι στοιχεία του πεδίου $GF(2^m)$. Τα συστατικά αυτά μπορούν να υπολογιστούν από το **r**(X) ως εξής :διαιρούμε το **r**(X) με το ελάχιστο πολυώνυμο $\varphi_i(X)$ του a^i και παίρνουμε :

r(X) = $a_i(X)\ \varphi_i(X)$ + $b_i(X)$ όπου $b_i(X)$ είναι το υπόλοιπο με βαθμό μικρότερο του $\varphi_i(X)$.Από τη στιγμή που $\varphi_i(a^i) = 0$, έχουμε

$S_i = r(a^i) = b_i(a^i)$ (6.14)

Επομένως το συστατικό του συνδρόμου S_i το παίρνουμε αν αντικαταστήσουμε στο $b_i(X)$ όπου

$X = a^i$. Από τη στιγμή που τα $\alpha,\alpha^2,...,\alpha^{2t}$ είναι οι ρίζες του κάθε κωδικού πολυωνύμου , $v(a^i)=0$ για $0 \le i \le 2t$. Προκύπτει από τις (6.11) και (6.13) ότι η σχέση του συνδρόμου και του error pattern θα είναι $S_i = e(a^i)$ (6.15)

για $0 \le i \le 2t$ από την οποία φαίνεται ότι το σύνδρομο S εξαρτάται μόνο από το error pattern. Υποθέτουμε ότι το error pattern e(X) έχει *ν* λάθη στις θέσεις $X^{j1},X^{j2},...,X^{jv}$ δηλαδή

$e(X) = X^{j1} + X^{j2} +...+ X^{jv}$ (6.16)

όπου $0 \le j1<j2<...<jv <n$. Από τις (6.15) και (6.16) προκύπτει το ακόλουθο σετ εξισώσεων:

$S_1 = a^{j1} + a^{j2} + \ldots + a^{jv}$

$S_2 = (a^{j1})^2 + (a^{j2})^2 + \ldots + (a^{jv})^2$ (6.17)

$S_3 = (a^{j1})^3 + (a^{j2})^3 + \ldots + (a^{jv})^3$

.

.

$S_{2t} = (a^{j1})^{2t} + (a^{j2})^{2t} + \ldots + (a^{jv})^{2t}$

Τα $a^{j1}, a^{j2}, \ldots, a^{jv}$ είναι άγνωστα .Κάθε μέθοδος επίλυσης αυτών των εξισώσεων είναι μέθοδος αποκωδικοποίησης για τους BCH κώδικες . Όταν προσδιοριστούν τα $a^{j1}, a^{j2}, \ldots, a^{jv}$ οι δυνάμεις $j^1, j^2, \ldots, j^v$ θα υποδεικνύουν τις θέσεις λάθους του e(X)της σχέσης (6.16). Γενικά οι εξισώσεις

των σχέσεων (6.17) έχουν πολλές πιθανές λύσεις (2^{κ}) και κάθε λύση μας δίνει διαφορετικό error pattern. Αν ο αριθμός των λαθών του πραγματικού error pattern e(X) είναι t ή λιγότερα η λύση που δίνει το error pattern με το μικρότερο αριθμό λαθών είναι η σωστή λύση. Αυτό σημαίνει ότι το error pattern που αντιστοιχεί σε αυτή τη λύση είναι το πιο πιθανό error pattern e(X) που έχει προκληθεί από το θόρυβο του καναλιού. Για μεγάλο t λύνοντας τις εξισώσεις των σχέσεων (6.17) είναι κάτι δύσκολο και αναποτελεσματικό. Στη συνέχεια περιγράφεται ένας αποτελεσματικός τρόπος που καθορίζει τα $a^{j\lambda}$ για $\lambda = 1,2,..\nu$ από τα συστατικά των συνδρόμων S_i.
Για ευκολία θεωρούμε $\beta_\lambda = \alpha^{j\lambda}$ (6.18)

για $1 \leq \lambda \leq \nu$.

Τα στοιχεία αυτά τα αποκαλούμε αριθμούς θέσης των λαθών (error location numbers) από τη στιγμή που μας υποδεικνύουν τις θέσεις στις οποίες βρίσκονται τα λάθη. Οι εξισώσεις της (6.17) μπορούν να εκφραστούν ως ακολούθως :

$S_1 = \beta_1 + \beta_2 + \ldots + \beta_\nu$

$S_2 = (\beta_1)^2 + (\beta_2)^2 + \ldots + (\beta_\nu)^2$ (6.19)

.

.

$S_{2t} = (\beta_1)^{2t} + (\beta_2)^{2t} + \ldots + (\beta_\nu)^{2t}$

Αυτές οι 2t εξισώσεις είναι συμμετρικές συναρτήσεις των $\beta_1, \beta_2, \ldots, \beta_\nu$ και είναι γνωστές ως συμμετρικές συναρτήσεις αθροίσματος δυνάμεων(power – sum symmetric functions).Στο σημείο αυτό ορίζεται το ακόλουθο πολυώνυμο :

$\sigma(X) = (1+\beta_1 X)(1+\beta_2 X)..... (1+\beta_\nu X) = \sigma_o + \sigma_1 X + \ldots + \sigma_\nu X^\nu$ (6.20)

Οι ρίζες του σ(X) δηλαδή τα $(\beta_1)^{-1}, (\beta_2)^{-1}, \ldots, (\beta_\nu)^{-1}$ είναι οι αντίστροφοι των αριθμών θέσης των λαθών. Για αυτό το λόγο το σ(X) καλείται πολυώνυμο θέσης λαθών. Και πάλι το σ(X) είναι ένα άγνωστο πολυώνυμο που πρέπει να καθοριστούν οι συντελεστές του. Οι συντελεστές του σ(X) και οι αριθμοί θέσης των λαθών σχετίζονται με τις ακόλουθες εξισώσεις.:

$\sigma_0=1$

$\sigma_1= \beta_1 +\beta_2+\ldots+ \beta_\nu$

$\sigma_2= \beta_1 \beta_2 +\beta_2 \beta_3+\ldots+ \beta_{\nu-1} \beta_\nu$ (6.21)

.

.

$\sigma_\nu= \beta_1\beta_2\ldots \beta_\nu$

Τα σ_i είναι γνωστά ως βασικές συμμετρικές συναρτήσεις των β_λ. Από τις σχέσεις (6.19) και (6.21)βλέπουμε ότι τα σ_i σχετίζονται με τα S_i. Σχετίζονται με τα συστατικά του συνδρόμου με τις ακόλουθες ταυτότητες του Νεύτωνα :

$S_1 +\sigma_1 =0$

$S_2+ \sigma_1S_1+ 2\sigma_2 = 0$

$S_3 +\sigma_1S_2+\sigma_2S_1+3\sigma_3=0$

. (6.22)

.

.

$S_\nu +\sigma_1S_{\nu-1}+...+\sigma_{\nu-1}S_1+\nu\sigma_\nu=0$

$S_{\nu+1} +\sigma_1S_\nu+...+\sigma_{\nu-1}S_2+\nu S_1=0$

Στη δυαδική περίπτωση όπου 1+1=2=0 έχουμε

$i\sigma_i = \begin{cases} \sigma_i \gamma\iota\alpha\, i\mu o\nu\acute{o}i \\ 0 \gamma\iota\alpha\, i\zeta\upsilon\gamma\acute{o}i \end{cases}$

Αν είναι δυνατόν να καθοριστούν τα σ_o , σ_1 ,…, σ_ν από τις εξισώσεις (6.22) οι αριθμοί θέσης των λαθών β_1 ,β_2,…, β_ν βρίσκονται αν από τις ρίζες του σ(X) ,το πολυώνυμο θέσης λαθών. Και σε αυτήν την περίπτωση οι λύσεις των εξισώσεων (6.22) είναι πολλές αλλά εμείς επιλέγουμε την λύση εκείνη για την οποία το πολυώνυμο σ(X) είναι ελάχιστου βαθμού. Αυτό το πολυώνυμο σ(X) θα παρήγαγε ένα error pattern με το μικρότερο αριθμό λαθών. Αν $\nu \leq t$ αυτό το σ(X) θα μας έδινε το πραγματικό error pattern e(X) . Στη συνέχεια περιγράφονται τα βήματα της διαδικασίας εκείνης που καθορίζει το πολυώνυμο σ(X) ελάχιστου βαθμού που ικανοποιεί τις πρώτες 2t εξισώσεις των (6.22) που ισχύει γενικά για τους BCH κώδικες.

Βήμα 1 Υπολογίζουμε το σύνδρομο $S = (S_1, S_2,…, S_{2t})$ από το λαμβανόμενο πολυώνυμο r(X).

Βήμα2 Καθορίζουμε το πολυώνυμο σ(X) θέσης λαθών από τα συστατικά του συνδρόμου S_1, S_2,…, S_{2t}.

Βήμα 3 Καθορίζουμε τους αριθμούς θέσης λαθών β_1 ,β_2,…, β_ν από τις ρίζες του πολυωνύμου σ(X) και διορθώνουμε τα λάθη του r(X).

Τα βήματα 1 και 3 είναι πολύ απλά ,το βήμα 2 είναι το πιο πολύπλοκο τμήμα της αποκωδικοποίησης σε ένα BCH κώδικα.

Επαναληπτικός αλγόριθμος για την εύρεση του πολυωνύμου θέσης λαθών σ(X)

Το πρώτο βήμα της επανάληψης είναι η εύρεση του πολυωνύμου ελάχιστου βαθμού $\sigma^{(1)}(X)$ του οποίου οι συντελεστές ικανοποιούν την πρώτη ταυτότητα του Νεύτωνα της (6.22).Το επόμενο βήμα είναι να **ελέγξουμε** αν οι συντελεστές του $\sigma^{(1)}(X)$ ικανοποιούν και τη δεύτερη ταυτότητα του Νεύτωνα της (6.22). Αν οι συντελεστές του $\sigma^{(1)}(X)$ όντως ικανοποιούν τη δεύτερη ταυτότητα του Νεύτωνα της (6.22) θέτουμε $\sigma^{(2)}(X)= \sigma^{(1)}(X)$. Αν οι συντελεστές του $\sigma^{(1)}(X)$ δεν ικανοποιούν τη δεύτερη ταυτότητα του Νεύτωνα της (6.22) ένας όρος διόρθωσης προστίθεται στο $\sigma^{(1)}(X)$ για να διαμορφώσει το $\sigma^{(2)}(X)$ έτσι ώστε το $\sigma^{(2)}(X)$ να έχει τον ελάχιστο βαθμό και οι συντελεστές του να ικανοποιούν τις πρώτες δύο ταυτότητες του Νεύτωνα της (6.22).Το τρίτο βήμα της επανάληψης είναι η εύρεση του ελάχιστου βαθμού πολυωνύμου $\sigma^{(3)}(X)$από το $\sigma^{(2)}(X)$ τέτοιο ώστε οι συντελεστές του $\sigma^{(3)}(X)$ να ικανοποιούν τις τρεις πρώτες ταυτότητες του Νεύτωνα της (6.22).Ξανά ελέγχουμε αν οι συντελεστές του $\sigma^{(2)}(X)$ ικανοποιούν και τη τρίτη ταυτότητα του Νεύτωνα της (6.22).Αν αυτό συμβαίνει τότε θέτουμε $\sigma^{(3)}(X)= \sigma^{(2)}(X)$ ενώ στην αντίθετη περίπτωση ένας όρος

διόρθωσης προστίθεται στο $\sigma^{(2)}(X)$ για να διαμορφώσει το $\sigma^{(3)}(X)$.Η επανάληψη συνεχίζεται έως ότου βρούμε το $\sigma^{(2t)}(X)$. Τότε το $\sigma^{(2t)}(X)$ θεωρείται ότι είναι το πολυώνυμο θέσης λαθών σ(X) δηλαδή , $\sigma(X)= \sigma^{(2t)}(X)$.

Αυτό το σ(X) θα αποδώσει και το error pattern e(X) ελάχιστου βάρους που θα ικανοποιεί και τις εξισώσεις (6.17).Αν ο αριθμός των λαθών του λαμβανόμενου πολυωνύμου r(X) είναι t ή μικρότερος τότε το σ(X) παράγει το πραγματικό error pattern.

Έστω $\sigma(X) = 1 + \sigma_1^{(\mu)}X +\ldots+ \sigma_{\lambda\mu}{}^{(\mu)}X^{\lambda\mu}$ (6.23)

είναι το ελάχιστου βαθμού πολυώνυμο που έχει καθοριστεί στο μ βήμα επανάληψης και του οποίου οι συντελεστές ικανοποιούν τις πρώτες μ ταυτότητες του Νεύτωνα της (6.22).Για να καθοριστεί το $\sigma^{(\mu+1)}(X)$ υπολογίζουμε την παρακάτω ποσότητα:

$d_\mu = S_{\mu+1} + \sigma_1^{(\mu)} S_\mu + \sigma_2^{(\mu)} S_{\mu-1} +\ldots+ \sigma_{\lambda\mu}{}^{(\mu)} S_{\mu+1-\lambda\mu}$ (6.24)

Αυτή η ποσότητα d_μ ονομάζεται μ $_{οστή}$ διαφορά. Αν $d_\mu =0$ οι συντελεστές του $\sigma^{(\mu)}(X)$ ικανοποιούν την (μ+1) ταυτότητα του Νεύτωνα. Υπό αυτές τις συνθήκες θέτουμε $\sigma^{(\mu+1)}(X)= \sigma^{(\mu)}(X)$.

Αν το $d_\mu \neq 0$ τότε οι συντελεστές του $\sigma^{(\mu)}(X)$ δεν ικανοποιούν την (μ+1) ταυτότητα του Νεύτωνα και ένας όρος διόρθωσης θα πρέπει να προστεθεί στο $\sigma^{(\mu)}(X)$ για να πάρουμε το $\sigma^{(\mu+1)}(X)$.Για να πετύχουμε αυτή τη διόρθωση επιστρέφουμε πίσω πριν από το μ βήμα για να καθορίσουμε ένα πολυώνυμο $\sigma^{(\rho)}(X)$ τέτοιο ώστε η ρ διαφορά $d_\rho \neq 0$ και $\rho - \lambda_\rho$ έχει τη μέγιστη τιμή. Τότε

$\sigma^{(\mu+1)}(X)= \sigma^{(\mu)}(X) + d_\mu d_r^{-1} X^{(\mu-\rho)} \sigma^{(\rho)}(X)$, (6.25)

που είναι το ελάχιστου βαθμού πολυώνυμο του οποίου οι συντελεστές ικανοποιούν τις πρώτες (μ+1) ταυτότητες του Νεύτωνα. Για να υλοποιήσουμε την επανάληψη για την εύρεση του σ(X) ξεκινάμε με τον πίνακα (6.5)

Πίνακας 6.5

μ	$\sigma^{(\mu)}(X)$	d_μ	l_μ	$\mu - l_\mu$

-1	1	1	0	- 1
0	1	S_1	0	0
1				
2				
.				
.				
.2t				

και προχωράμε συμπληρώνοντας τον πίνακα, όπου λ_μ είναι ο βαθμός του $\sigma^{(\mu)}(X)$. Υποθέτοντας ότι έχουμε συμπληρώσει όλες τις γραμμές μέχρι την μ γραμμή συμπληρώνουμε την (μ+1) γραμμή ως ακολούθως :

1. Αν $d_\mu = 0$ τότε $\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X)$ και $\lambda_{\mu+1} = \lambda_\mu$
2. Αν $d_\mu \neq 0$ βρίσκουμε μια άλλη προηγούμενη γραμμή της μ τη γραμμή ρ για την οποία $d_\rho \neq 0$ και ο αριθμός $\rho - \lambda_\rho$ στην τελευταία στήλη του πίνακα έχει την μεγαλύτερη τιμή. Τότε το

   $\sigma^{(\mu+1)}(X)$ δίνεται από τη σχέση (6.25)και

   $\lambda_{\mu+1} = \max(\lambda_\mu , \lambda_\rho+\mu-\rho)$(6.26)

Σε κάθε περίπτωση

$$d_{\mu+1} = S_{\mu+2} + \sigma_1^{(\mu+1)} S_{\mu+1} + \sigma_2^{(\mu)} S_{\mu-1} + \ldots + \sigma_{\lambda\mu+1}^{(\mu+1)} S_{\mu+2-\lambda\mu} \quad (6.27)$$

όπου $\sigma_i^{(\mu+1)}$ είναι οι συντελεστές του $\sigma^{(\mu+1)}(X)$.Το πολυώνυμο $\sigma^{(2t)}(X)$ που βρίσκεται στη τελευταία γραμμή θα πρέπει να είναι το απαιτούμενο σ(X). Αν έχει βαθμό μεγαλύτερο του t , υπάρχουν περισσότερα από t λάθη στο λαμβανόμενο πολυώνυμο r(X) και γενικά δεν είναι δυνατόν να τα εντοπίσουμε.

Αν ο αριθμός των λαθών σε ένα λαμβανόμενο πολυώνυμο r(X) είναι μικρότερος από την σχεδιαστική ικανότητα διόρθωσης λαθών t του κώδικα , δεν είναι αναγκαίο να διεξάγουμε τα 2t βήματα της επανάληψης για να βρούμε το πολυώνυμο θέσης λαθών σ(X). Έστω $\sigma^{(\mu)}(X)$ και d_μ είναι οι λύσεις και η διαφορά που παίρνουμε στο μ βήμα της επανάληψης. Έστω λ_μ ο βαθμός του σ

$^{(\mu)}$ (X). Έχει αποδειχτεί ότι αν d_μ και οι διαφορές στα επόμενα t - $\lambda_\mu - 1$ βήματα είναι όλα μηδενικά , σ $^{(\mu)}$ (X) είναι το πολυώνυμο θέσης λαθών. Βασισμένοι σε αυτό το γεγονός αν ο αριθμός των λαθών του λαμβανόμενου πολυώνυμου r(X) είναι ν (ν $\leq$t) τότε μόνο t + ν βήματα επανάληψης απαιτούνται για να καθορίσουν το πολυώνυμο θέσης λαθών σ(X). Αν το ν είναι μικρό η μείωση του αριθμού των βημάτων επανάληψης οδηγεί στη αύξηση της ταχύτητας της αποκωδικοποίησης. Αυτό ο επαναληπτικός αλγόριθμος για την εύρεση του σ(X) που περιγράφηκε παραπάνω δεν εφαρμόζεται μόνο στους δυαδικούς BCH κώδικες αλλά και στους μη δυαδικούς BCH κώδικες.

Απλοποιημένος αλγόριθμος για την εύρεση του σ(X)

Για τους δυαδικούς BCH κώδικες δεν είναι αναγκαία η συμπλήρωση των κενών 2t γραμμών του πίνακα 6.5 για την εύρεση του σ(X). Ένας απλοποιημένος αλγόριθμος μπορεί να χρησιμοποιηθεί ο οποίος απαιτεί τη συμπλήρωση ενός πίνακα μόνο t κενών γραμμών. Ένας τέτοιος πίνακας είναι και ο 6.6.

Πίνακας 6.6

μ	$\sigma^{(\mu)}(X)$	d_μ	l_μ	$2\mu - l_\mu$
-1/2	1	1	0	- 1
0	1	S_1	0	0
1				
2				
.				
.				
.				
t				

Υποθέτουμε ότι έχουμε συμπληρώσει όλες τις γραμμές μέχρι και τη γραμμή μ , συμπληρώνουμε την γραμμή (μ+1) ως ακολούθως :

1. Αν $d_\mu = 0$ τότε $\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X)$

2. Αν $d_\mu \neq 0$ βρίσκουμε μια άλλη προηγούμενη γραμμή της μ τη γραμμή ρ για την οποία $d_\rho \neq 0$ και ο αριθμός $2\rho - \lambda_\rho$ στην τελευταία στήλη του πίνακα έχει την μεγαλύτερη δυνατή τιμή. Τότε το

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X) + d_\mu d_\rho^{-1} X^{2(\mu-\rho)} \sigma^{(\rho)}(X), \qquad (6.28)$$

Σε κάθε περίπτωση $\lambda_{\mu+1}$ είναι ακριβώς ο βαθμός του $\sigma^{(\mu+1)}(X)$,και η διαφορά στο (μ+1) βήμα είναι :

$$d_{\mu+1} = S_{2\mu+3} + \sigma_1^{(\mu+1)} S_{2\mu+2} + \sigma_2^{(\mu+1)} S_{2\mu+1} + \ldots + \sigma_{\lambda\mu+1}^{(\mu+1)} S_{2\mu+3-\lambda\mu+1} \qquad (6.29)$$

Το πολυώνυμο $\sigma^{(t)}(X)$ που βρίσκεται στη τελευταία γραμμή πρέπει να είναι το απαιτούμενο σ(X). Αν έχει βαθμό μεγαλύτερο του t , υπάρχουν περισσότερα από t λάθη στο λαμβανόμενο πολυώνυμο r(X) και γενικά δεν είναι δυνατόν να τα εντοπίσουμε. Οι υπολογισμοί που απαιτούνται στον απλοποιημένο αλγόριθμο είναι ακριβώς οι μισοί των υπολογισμών που απαιτούνται για τον γενικό αλγόριθμο. Πρέπει όμως να λάβουμε υπόψη μας ότι ο απλοποιημένος αλγόριθμος εφαρμόζεται μόνο στους δυαδικούς BCH κώδικες.

Αν ο αριθμός των λαθών σε ένα λαμβανόμενο πολυώνυμο r(X) είναι μικρότερος από την σχεδιαστική ικανότητα διόρθωσης λαθών t του κώδικα , δεν είναι αναγκαίο να διεξάγουμε τα t βήματα της επανάληψης για να βρούμε το πολυώνυμο θέσης λαθών σ(X) για ένα δυαδικό BCH κώδικα. Αν για κάποια μ, d_μ και οι διαφορές στα επόμενα $[(t - \lambda_\mu - 1)/2]$ βήματα επανάληψης είναι μηδέν τότε το $\sigma^{(\mu)}(X)$ είναι το πολυώνυμο θέσης λαθών. Αν ο αριθμός των λαθών του λαμβανόμενου πολυώνυμου r(X) είναι ν ($\nu \leq t$) τότε μόνο$[(t + \nu)/2]$βήματα επανάληψης απαιτούνται για να καθορίσουν το πολυώνυμο θέσης λαθών σ(X).

Εύρεση των αριθμών θέσης λάθους και διόρθωση λαθών

Το τελευταίο βήμα στην αποκωδικοποίηση ενός BCH κώδικα είναι οι εύρεση των αριθμών που μας δείχνουν τις θέσεις που βρίσκονται τα λάθη οι οποίοι είναι οι αντίστροφοι των ριζών του σ(X). Οι ρίζες του σ(X) μπορούν απλά να βρεθούν αντικαθιστώντας $1,\alpha,\alpha^2,\ldots,\alpha^{n-1}$ ($n=2^m - 1$)στο σ(X).Από

τη στιγμή που το $\alpha^n = 1$, $\alpha^{-\lambda} = \alpha^{n-\lambda}$.Επομένως αν α^λ είναι ρίζα του σ(X) , $\alpha^{n-\lambda}$ είναι ένας αριθμός της θέσης λάθους και το r_{n-1} λαμβανόμενο ψηφίο είναι ένα λανθασμένο ψηφίο. Μια διαδικασία που έχει αναπτυχθεί για την ανακάλυψη των αριθμών των θέσεων λάθους περιγράφεται σε αυτό το σημείο. Το λαμβανόμενο διάνυσμα

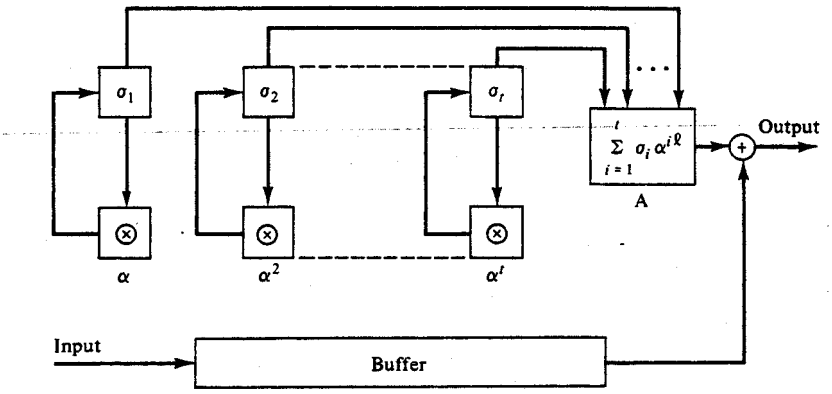
$$\mathbf{r}(X) = r_o + r_1X + \ldots + r_{n-1}X^{n-1}$$

αποκωδικοποιείται σε μία βάση bit by bit. Τα ψηφία υψηλότερης τάξης αποκωδικοποιούνται πρώτα. Για να αποκωδικοποιήσουμε το r_{n-1} , ο αποκωδικοποιητής ελέγχει αν το α^{n-1} είναι αριθμός θέσης λάθους που είναι ισοδύναμο με τον έλεγχο για το αν ο αντίστροφος του ο α είναι ρίζα του σ(X). Αν α είναι ρίζα τότε

$$\sigma(a^i) = 1 + \sigma_1\alpha^i + \ldots + \sigma_\nu\alpha^{\nu i} = 0$$

Επομένως για να αποκωδικοποιήσει το r_{n-1},ο αποκωδικοποιητής διαμορφώνει τα $\sigma_1\alpha^i, \ldots, \sigma_\nu\alpha^{\nu i}$. Αν το άθροισμα $1 + \sigma_1\alpha^i + \ldots + \sigma_\nu\alpha^{\nu i} = 0$ τότε το α^{n-1} είναι λανθασμένο ψηφίο, διαφορετικά το r_{n-1} είναι σωστό ψηφίο. Για την αποκωδικοποίηση του $r_{n-\lambda}$,ο αποκωδικοποιητής διαμορφώνει τα $\sigma_1\alpha^\lambda, \ldots, \sigma_\nu\alpha^{\nu\lambda}$ και ελέγχει τα αθροίσματα $1 + \sigma_1\alpha^\lambda + \sigma_2\alpha^{2\lambda} + \ldots + \sigma_\nu\alpha^{\nu\lambda}$.

Αν το άθροισμα είναι μηδέν τότε το α^λ είναι ρίζα του σ(X) και το $r_{n-\lambda}$ είναι λανθασμένο ψηφίο διαφορετικά είναι σωστό. Αυτή η διαδικασία ελέγχου που περιγράφηκε παραπάνω μπορεί να υλοποιηθεί με το κύκλωμα που φαίνεται στο παρακάτω **σχήμα 6.1**



Σχήμα 6.1 Μονάδα ανίχνευσης κυκλικών θέσεων λαθών

Οι t σ- καταχωρητές αποθηκεύονται αρχικά με τα $\sigma_1,\sigma_2,\sigma_3,...,\sigma_t$ που έχουν υπολογιστεί στο 2^o βήμα της αποκωδικοποίησης . Αμέσως πριν διαβαστεί το r_{n-1},από τον buffer οι t πολλαπλασιαστές πάλλονται μία φορά. Εκτελούνται οι πολλαπλασιασμοί και τα $\sigma_1\alpha$, $\sigma_2\alpha^2$,…, $\sigma_\nu\alpha^\nu$ αποθηκεύονται στους σ καταχωρητές. Η έξοδος του λογικού κυκλώματος Α είναι 1 αν και μόνο αν το άθροισμα $1+\sigma_1\alpha+\sigma_2\alpha^2+\ldots+\sigma_\nu\alpha^\nu=0$ διαφορετικά η έξοδος θα είναι 0.Το ψηφίο r_{n-1},διαβάζεται από τον buffer και διορθώνεται από την έξοδο του Α. Έχοντας αποκωδικοποιήσει το ψηφίο r_{n-1}, οι t πολλαπλασιαστές πάλλονται μία ακόμη φορά. Αυτή τη φορά στους σ-καταχωρητές αποθηκεύονται τα $\sigma_1\alpha^2$, $\sigma_2\alpha^4$,…, $\sigma_\nu\alpha^{2\nu}$.Το άθροισμα $1+\sigma_1\alpha^2+\sigma_2\alpha^4+\ldots+\sigma_\nu\alpha^{2\nu}$ ελέγχεται για το αν είναι ίσο με μηδέν. Το ψηφίο r_{n-2},διαβάζεται από τον buffer και διορθώνεται με τον ίδιο τρόπο που διορθώθηκε το ψηφίο r_{n-1} .Η διαδικασία αυτή συνεχίζεται έως ότου διαβαστεί όλο το λαμβανόμενο διάνυσμα από το buffer.

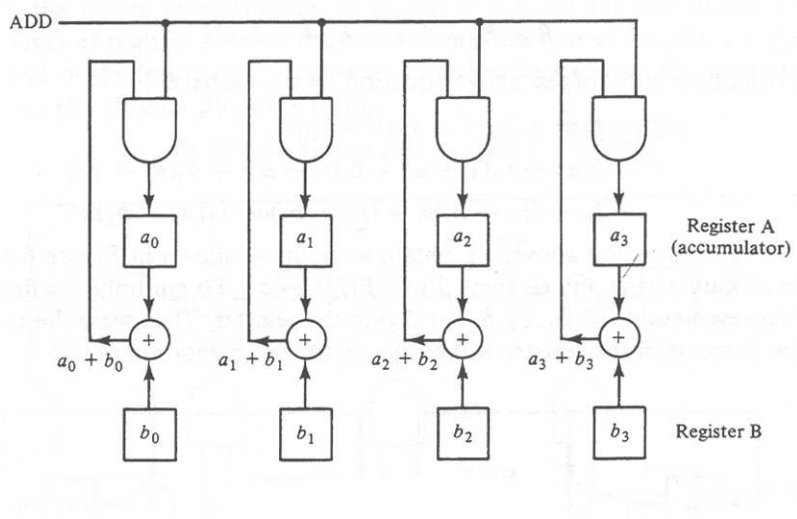
Ο παραπάνω αλγόριθμος αποκωδικοποίησης εφαρμόζεται και σε μη βασικοί (nonprimitive) BCH κώδικες. Τα 2t συστατικά του συνδρόμου δίνονται από $S_i = r(\beta^i)$ για $1 \le i \le 2t$.

6.3Υλοποίηση της αριθμητικής των Galois πεδίων

Από όσα έχουμε δει έως τώρα η αποκωδικοποίηση των BCH κωδίκων απαιτεί υπολογισμούς χρησιμοποιώντας την αριθμητική των Galois πεδίων. Η αριθμητική των Galois πεδίων μπορεί να υλοποιηθεί ευκολότερα από την συνήθη γιατί δεν έχει κρατούμενα. Στην παράγραφο αυτή παρουσιάζονται τα κυκλώματα εκείνα που εκτελούν πρόσθεση και πολλαπλασιασμό σε ένα Galois πεδίο. Για χάρη απλότητας θεωρούμε την αριθμητική στο Galois πεδίοGF(2^4)που δίνεται στον πίνακα 1 του κεφαλαίου 2.

Για να προσθέσουμε δύο στοιχεία του πεδίου απλά προσθέτουμε τις διανυσματικές αναπαραστάσεις τους . Το εξαγόμενο διάνυσμα είναι η διανυσματική αναπαράσταση του αθροίσματος των δύο στοιχείων του πεδίου. Έστω για παράδειγμα ότι θέλουμε να προσθέσουμε τα στοιχεία α^7 και α^{13}του GF(2^4). Οι διανυσματικές αναπαραστάσεις των δύο στοιχείων θα είναι βάση του πίνακα 2.8 (1101)και (1011) αντίστοιχα. Το διανυσματικό τους άθροισμα θα είναι (1101)+ (1011)= (0110) που είναι η διανυσματική αναπαράσταση του α^5.

Η πρόσθεση δύο στοιχείων του πεδίου μπορεί να πραγματοποιηθεί με τη χρήση του κυκλώματος που φαίνεται στο σχήμα 6.2 Αρχικά οι δύο διανυσματικές αναπαραστάσεις των δύο στοιχείων που πρόκειται να προστεθούν φορτώνονται στους δύο καταχωρητές Α και Β. Το διανυσματικό τους άθροισμα εμφανίζεται στην είσοδο του καταχωρητή Α. Όταν ο καταχωρητής Α πάλλεται το άθροισμα φορτώνεται στον καταχωρητή Α.(ο καταχωρητής Α παίζει το ρόλο συσσωρευτή).



Σχήμα 6.2 Αθροιστής Galois πεδίων

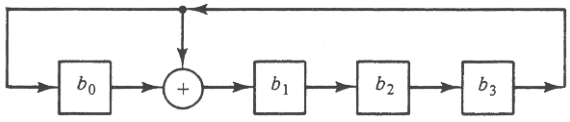
Για τον πολλαπλασιασμό θεωρούμε πρώτα την περίπτωση που το στοιχείο του πεδίου πολλαπλασιάζεται με ένα σταθερό στοιχείο από το ίδιο πεδίο. Υποθέτουμε ότι θέλουμε να πολλαπλασιάσουμε ένα στοιχείο του πεδίου β του GF(2^4) με το βασικό (primitive)στοιχείο α του οποίου το ελάχιστο πολυώνυμο είναι φ(X)= 1+ X + X^4. Το στοιχείο β μπορεί να εκφραστεί ως πολυώνυμο του α ως εξής :

$\beta = b_0+b_1\alpha+b_2\alpha^2+b_3\alpha^3$.

Πολλαπλασιάζοντας και τα δύο μέρη με α και χρησιμοποιώντας το γεγονός ότι $\alpha^4=1+\alpha$, παίρνουμε την ακόλουθη ισότητα:

$\alpha\beta= b_3+(b_0+ b_3)\alpha+b_1\alpha^2+b_2\alpha^3$.

Ο πολλαπλασιασμός αυτός μπορεί να υλοποιηθεί από το κύκλωμα του καταχωρητή ολίσθησης με ανάδραση που φαίνεται στο **σχήμα 6.3.**



Σχήμα 6.3 κύκλωμα πολλαπλασιασμού ενός αυθαίρετου στοιχείου του GF(2^4) με α

Αρχικά η διανυσματική αναπαράσταση (b_0,b_1,b_2,b_3) του β φορτώνεται στον καταχωρητή. Κατόπιν ο καταχωρητής πάλλεται. Τα νέα περιεχόμενα στον καταχωρητή διαμορφώνουν τη διανυσματική αναπαράσταση του αβ. Το κύκλωμα του σχήματος 6.3 μπορεί να χρησιμοποιηθεί και για την παραγωγή όλων των μη μηδενικών στοιχείων του GF(2^4). Αρχικά φορτώνουμε το (1 0 0 0) (που είναι η διανυσματική αναπαράσταση του $\alpha^0 = 1$) στον καταχωρητή. Επιτυχείς ολισθήσεις του καταχωρητή μπορούν να παράγουν τις διανυσματικές αναπαραστάσεις των δυνάμεων του α, με την ίδια σειρά που φαίνεται στον πίνακα 2.8.Στο τέλος της $15^{ης}$ ολίσθησης ο καταχωρητής θα περιέχει ξανά το (1 0 0 0).

Μία άλλη περίπτωση είναι ο πολλαπλασιασμός ενός αυθαίρετου στοιχείου β με το α^3. Για μια ακόμη φορά εκφράζουμε το β σε πολυωνυμική μορφή,

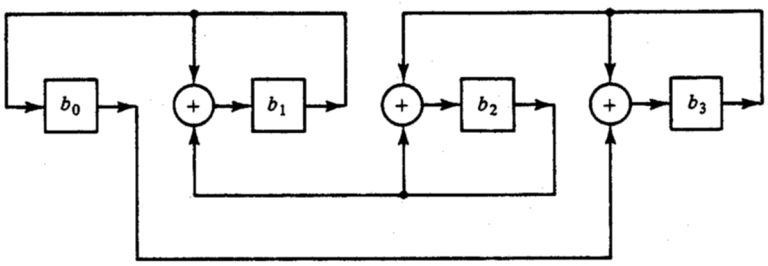
$\beta = b_0+b_1\alpha+b_2\alpha^2+b_3\alpha^3$.

Πολλαπλασιάζοντας και τα δύο μέρη της παραπάνω ισότητας με το α^3παίρνουμε

$\alpha^3\beta = b_0\,\alpha^3+b_1\alpha^4+b_2\,\alpha^5+b_3\alpha^6$

$= b_1+(b_1+ b_2)\alpha+(b_3+ b_2)\alpha^2+(b_0+ b_3)\alpha^3$.

Βασισμένοι στην παραπάνω σχέση χρησιμοποιούμε το κύκλωμα που φαίνεται στο σχήμα 6.4 που είναι σε θέση να πολλαπλασιάζει οποιοδήποτε στοιχείο β του GF(2^4) με το α^3.Για να γίνει ο πολλαπλασιασμός αρχικά φορτώνουμε τη διανυσματική αναπαράσταση (b_0,b_1,b_2,b_3) του β στον καταχωρητή. Στη συνέχεια πάλλεται ο καταχωρητής και τα νέα περιεχόμενα του καταχωρητή είναι η διανυσματική αναπαράσταση του $\alpha^3\beta$.



σχήμα6.4 κύκλωμα πολλαπλασιασμού ενός αυθαίρετου στοιχείου του $GF(2^4)$ με α^3

Στην επόμενη περίπτωση θεωρούμε τον πολλαπλασιασμό δύο αυθαίρετων στοιχείων του πεδίου.(Χρησιμοποιούμε ξανά την περίπτωση του $GF(2^4)$). Έστω δύο στοιχεία του $GF(2^4)$ που σε πολυωνυμική μορφή είναι :

$\beta = b_0+b_1\alpha+b_2\alpha^2+b_3\alpha^3$.

$\gamma = c_0+c_1\alpha+c_2\alpha^2+c_3\alpha^3$.

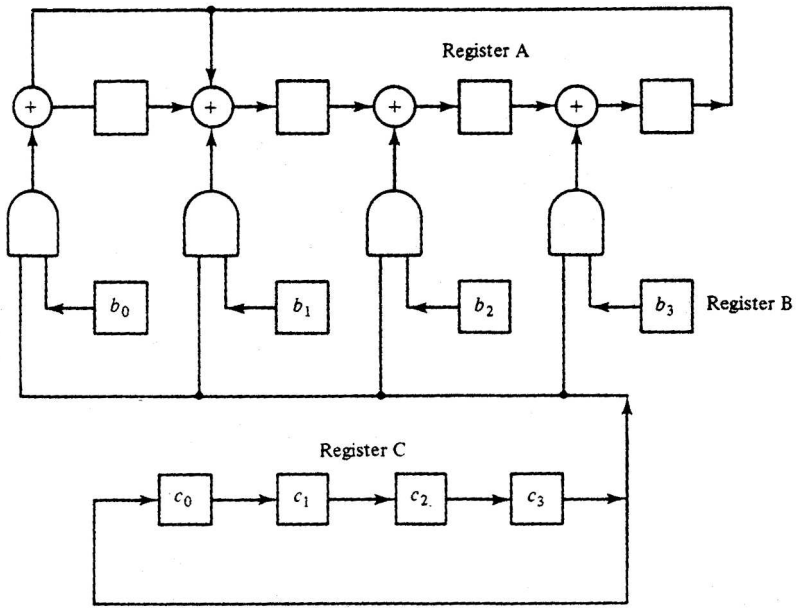
Το γινόμενο βγ μπορεί να εκφραστεί ως εξής :

$\beta\gamma =(((c_3\beta)\alpha+ c_2\beta)\alpha+ c_1\beta)\alpha + c_0\beta$ (6.30)

Το γινόμενο αυτό μπορεί να εξαχθεί ακολουθώντας τα εξής βήματα:

1. Πολλαπλασιάζουμε το $c_3\beta$ με το α και προσθέτουμε στο γινόμενο το $c_2\beta$.
2. Πολλαπλασιάζουμε το $(c_3\beta)\alpha+ c_2\beta$ με το α και προσθέτουμε στο γινόμενο το $c_1\beta$.
3. Πολλαπλασιάζουμε το $((c_3\beta)\alpha+ c_2\beta)\alpha+ c_1\beta$ με το α και προσθέτουμε στο γινόμενο το $c_0\beta$.

Ο πολλαπλασιασμός με το α μπορεί να γίνει και με το κύκλωμα του σχήματος 6.3. Αυτό το κύκλωμα μπορεί να τροποποιηθεί για να διεξάγει τους υπολογισμούς που δίνονται στη σχέση (6.30).Το κύκλωμα που προκύπτει φαίνεται στο **σχήμα 6.5**

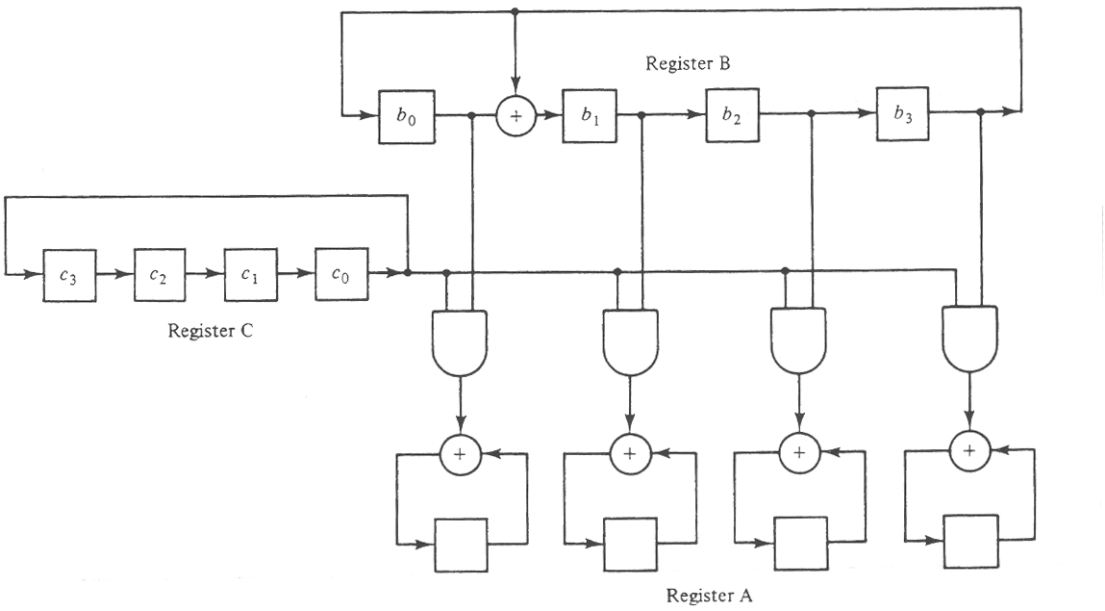


Σχήμα6.5 Κύκλωμα πολλαπλασιασμού δύο στοιχείων του GF(2^4)

Στην αρχή της λειτουργίας αυτού του κυκλώματος ο καταχωρητής ολίσθησης Α με ανάδραση είναι κενός και (b_0,b_1,b_2,b_3) και (c_0,c_1,c_2,c_3) οι διανυσματικές αναπαραστάσεις του β και του γ , φορτώνονται στους καταχωρητές Β και C αντίστοιχα. Οι καταχωρητές Α και C ολισθαίνουν 4 φορές. Στο τέλος της πρώτης ολίσθησης ο καταχωρητής Α περιέχει (c_3b_0, c_3b_1, c_3b_2, c_3b_3) που είναι η διανυσματική αναπαράσταση του c_3β. Στο τέλος της δεύτερης ολίσθησης ο καταχωρητής Α περιέχει τη διανυσματική αναπαράσταση του

(c_3β)α+ c_2β .Στο τέλος της τρίτης ολίσθησης τα περιεχόμενα του καταχωρητή Α είναι η διανυσματική αναπαράσταση του ((c_3β)α+ c_2β)α + c_1β. Στο τέλος της τέταρτης ολίσθησης τα περιεχόμενα του καταχωρητή Α είναι η διανυσματική αναπαράσταση του γινομένου βγ. Αν επιλέξουμε να εκφράσουμε το γινόμενο βγ = (((c0β) + c_1βα)+ c_2βα^2) + c_3β α^3(6.30)

το κύκλωμα που θα χρησιμοποιήσουμε είναι αυτό του **σχήματος 6.6**



Σχήμα 6.6 Κύκλωμα για τον πολλαπλασιασμό δύο στοιχείων του GF(2^4)

Τα β και γ , φορτώνονται στους καταχωρητές B και C αντίστοιχα και ο καταχωρητής ολίσθησης A με ανάδραση είναι αρχικά κενός. Στη συνέχεια οι καταχωρητές A και B και C ολισθαίνουν 4 φορές. Στο τέλος της τέταρτης ολίσθησης τα περιεχόμενα του καταχωρητή A είναι η διανυσματική αναπαράσταση του γινομένου βγ. Και τα δύο κυκλώματα πολλαπλασιασμού των **σχημάτων 6.5 και 6.6** έχουν την ίδια πολυπλοκότητα και απαιτούν τον ίδιο υπολογιστικό χρόνο.

Ο πολλαπλασιασμός δύο στοιχείων από το GF(2^4) μπορεί να υλοποιηθεί με ένα συνδυαστικό λογικό κύκλωμα με 2^m εισόδους και m εξόδους. Ο πλεονέκτημα αυτής της υλοποίησης είναι η ταχύτητα. Όμως για m>7 γίνεται πολύ πολύπλοκο και δαπανηρό. Ο πολλαπλασιασμός θα μπορούσε να προγραμματιστεί και από ένα γενικής χρήσης υπολογιστή ο οποίος θα απαιτούσε το ανώτερο 5m εκτελέσεις εντολών.

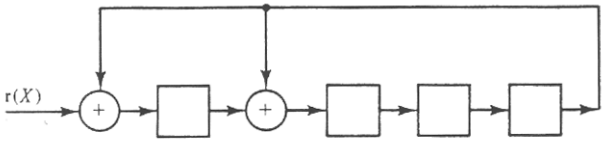
Έστω r(X) ένα πολυώνυμο στο GF(2). Ο τύπος υπολογισμού του r(X) είναι το πρώτο βήμα αποκωδικοποίησης ενός BCH κώδικα. Μπορεί να γίνει με ένα κύκλωμα πολλαπλασιασμού ενός στοιχείου του πεδίου με το α^i του GF(2^m). Και σε αυτή την περίπτωση χρησιμοποιούμε το GF(2^4).Υποθέτουμε ότι θέλουμε να υπολογίσουμε το

$\mathbf{r}(\alpha) = r_o + r_1\alpha + \ldots + r_{14}\alpha^{14}$ (6.31)

όπου το α είναι ένα βασικό στοιχείο του $GF(2^4)$. Το δεξί μέλος της (6.31) μπορεί να εκφραστεί

$\mathbf{r}(\alpha) = (...(((r_{14})\alpha + r_{13})\alpha + r_{12})\alpha + ...)\alpha + r_o$ (6.31)

Τότε ο υπολογισμός του **r**(α) επιτυγχάνεται με το να προσθέσουμε μία είσοδο στο κύκλωμα για τον πολλαπλασιασμό με το α όπως δείχνει το σχήμα 6.3 και το κύκλωμα που τελικώς χρησιμοποιείται είναι αυτό του **σχήματος 6.7.**



Σχήμα 6.7 Κύκλωμα υπολογισμού του r(a)

Στην αρχή της λειτουργίας αυτού του κυκλώματος ο καταχωρητής είναι κενός. Το διάνυσμα (r_o , r_1 ,…, r_{14})ολισθαίνει στο κύκλωμα ένα ψηφίο κάθε φορά. Μετά την πρώτη ολίσθηση ο καταχωρητής περιέχει το (r_{14},0,0,0), μετά τη δεύτερη ο καταχωρητής περιέχει τη διανυσματική αναπαράσταση $r_{14}\alpha + r_{13}$.Κατά τη ολοκλήρωση της τρίτης ολίσθησης ο καταχωρητής περιέχει τη διανυσματική αναπαράσταση $(r_{14}\alpha + r_{13})\alpha + r_{12}$.Όταν το τελευταίο ψηφίο r_o ολισθαίνει στο κύκλωμα ο καταχωρητής περιέχει το **r**(α) σε μορφή διανύσματος.

Στο σημείο αυτό εξετάζουμε τον υπολογισμό του $\mathbf{r}(\alpha^i)$. Έστω $\varphi_i(X)$ το ελάχιστο πολυώνυμο του α^i .Έστω ότι b(X) είναι το υπόλοιπο της διαίρεσης του r(X) με το $\varphi_i(X)$. Τότε το $\mathbf{r}(\alpha^i) = b(\alpha^i)$.Επομένως ο υπολογισμός του $\mathbf{r}(\alpha^i)$ είναι ισοδύναμος με τον υπολογισμό του $b(\alpha^i)$. Επομένως ο στόχος μας είναι η υλοποίηση κυκλώματος για τον υπολογισμό του $b(\alpha^i)$.Στο σημείο αυτό θεωρούμε ότι οι υπολογισμοί γίνονται στο $GF(2^4)$.Υποθέτουμε ότι έχουμε να υπολογίσουμε το $\mathbf{r}(\alpha^3)$.Το ελάχιστο πολυώνυμο του α^3 είναι $\varphi_3(X) = 1+X+X^2+X^3+X^4$. Το υπόλοιπο της διαίρεσης του r(X) με το $\varphi_3(X)$ έχει τη μορφή:

$b(X)= b_0+ b_1X+ b_2X^2+ b_3X^3$.

Τότε

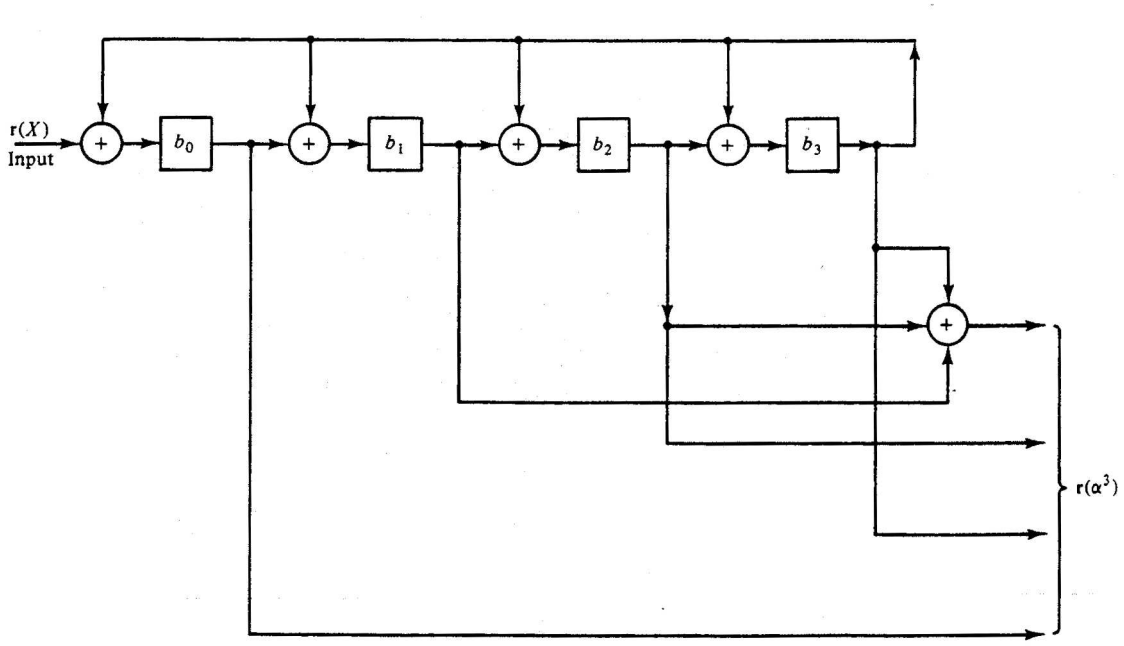
$\mathbf{r}(\alpha^3)= b(\alpha^3)$

$= b_0+ b_1 \alpha^3+ b_2 \alpha^6+ b_3 \alpha^9$

$= b_0+ b_1 \alpha^3+ b_2 (\alpha^2+ \alpha^3)+ b_3 (\alpha+ \alpha^3)$ (6.32)

$= b_0+ b_3 \alpha+ b_2 \alpha^2+(b_1 + b_2 + b_3)\alpha^3$

Το κύκλωμα του σχήματος **6.9**



Σχήμα 6.9 Κύκλωμα υπολογισμού του $\mathbf{r}(\alpha^3)$ στο $GF(2^4)$

Το κύκλωμα του σχήματος **6.9** υλοποιεί τον υπολογισμό του $\mathbf{r}(\alpha^3)$ βασισμένο στις σχέσεις (6.32)και η σύνδεση ανάδρασης του καταχωρητή ολίσθησης βασίζεται στο $\varphi_3(X)= 1+X+X^2+X^3+X^4$.Από τη στιγμή που το α^6 είναι συζυγής του α^3 θα έχει και το ίδιο ελάχιστο πολυώνυμο και επομένως και το $\mathbf{r}(\alpha^6)$ μπορεί να υπολογιστεί από το ίδιο υπόλοιπο b(X)που προκύπτει από τη διαίρεση του r(X) με $\varphi_3(X)$. Για τη διαμόρφωση του $\mathbf{r}(\alpha^6)$ οι συντελεστές του b(X) χρησιμοποιούνται ως εξής :

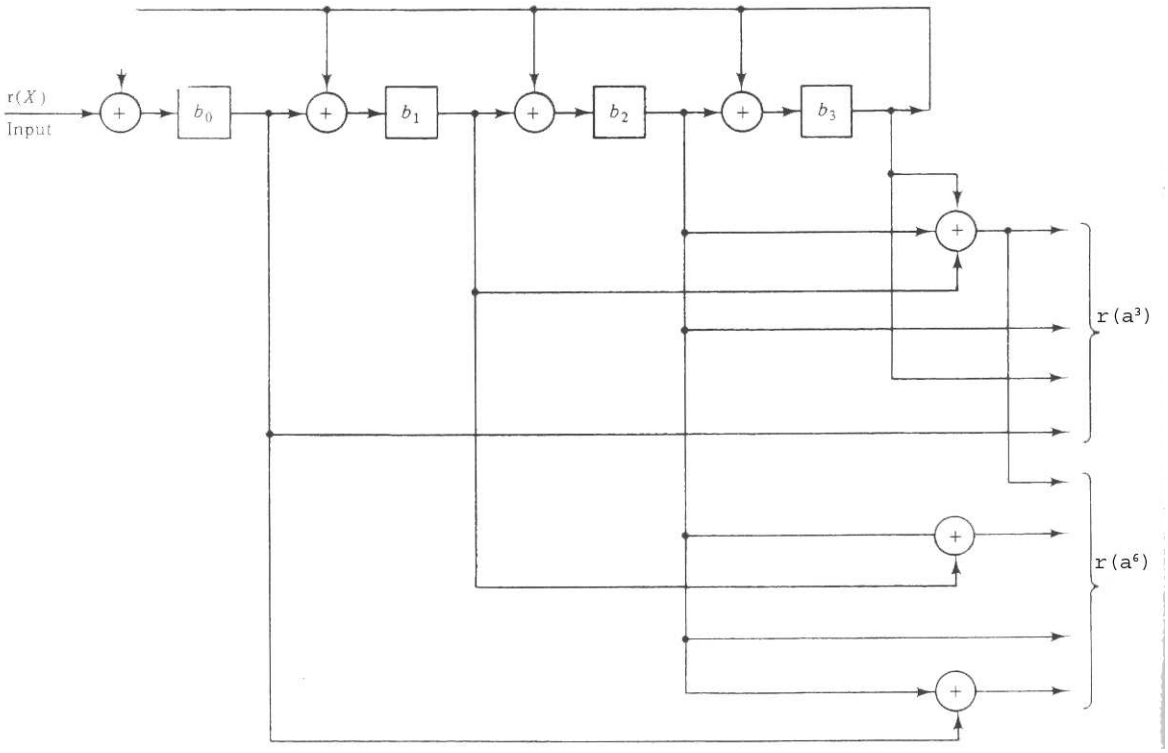
$\mathbf{r}(\alpha^6)= b(\alpha^6)$

$= b_0+ b_1 \alpha^6+ b_2 \alpha^{12}+ b_3 \alpha^{18}$

$= b_0+ b_1 (\alpha^2+ \alpha^3)+ b_2 (1+\alpha+\alpha^2+ \alpha^3)+ b_3 \alpha^3$

$= (b_0+ b_2) + b_2\alpha +(b_1+ b_2)\alpha 2+ (b_1+ b_2+ b_3)\alpha^3$

Το συνδυαστικό κύκλωμα για τον υπολογισμό του $\mathbf{r}(\alpha^3)$ και του $\mathbf{r}(\alpha^6)$ φαίνεται στο **σχήμα 6.10.**



Σχήμα 6.10

Η πράξη της διαίρεσης στο $GF(2^m)$ μπορεί να διεξαχθεί με το να υπολογιστεί αρχικά ο αντίστροφος του διαιρέτη β και μετά να υπολογιστεί το γινόμενο του με τον διαιρετέο.($\beta^{-1} = \beta^{2\wedge m-2}$).

6.4 Υλοποίηση της διόρθωσης λαθών

Κάθε βήμα της αποκωδικοποίησης ενός BCH κώδικα μπορεί να υλοποιηθεί είτε από ψηφιακό υλικό είτε με προγραμματισμό(σε ένα γενικής χρήσης υπολογιστή).Κάθε μία από τις δύο υλοποιήσεις έχει συγκεκριμένα πλεονεκτήματα.

Υπολογισμός συνδρόμου

Το πρώτο βήμα της αποκωδικοποίησης ενός t error correction BCH κώδικα είναι ο υπολογισμός των 2t συστατικών του συνδρόμου S_1, S_2,…, S_{2t}.Αυτά τα συστατικά του συνδρόμου μπορούν να ληφθούν με την αντικατάσταση των στοιχείων του πεδίου α^1, α^2,…, α^{2t} στο λαμβανόμενο πολυώνυμο r(X). Στην προγραμματιστική υλοποίηση η αντικατάσταση του α^i στο r(X) επιτυγχάνεται καλύτερα με τον ακόλουθο τρόπο:

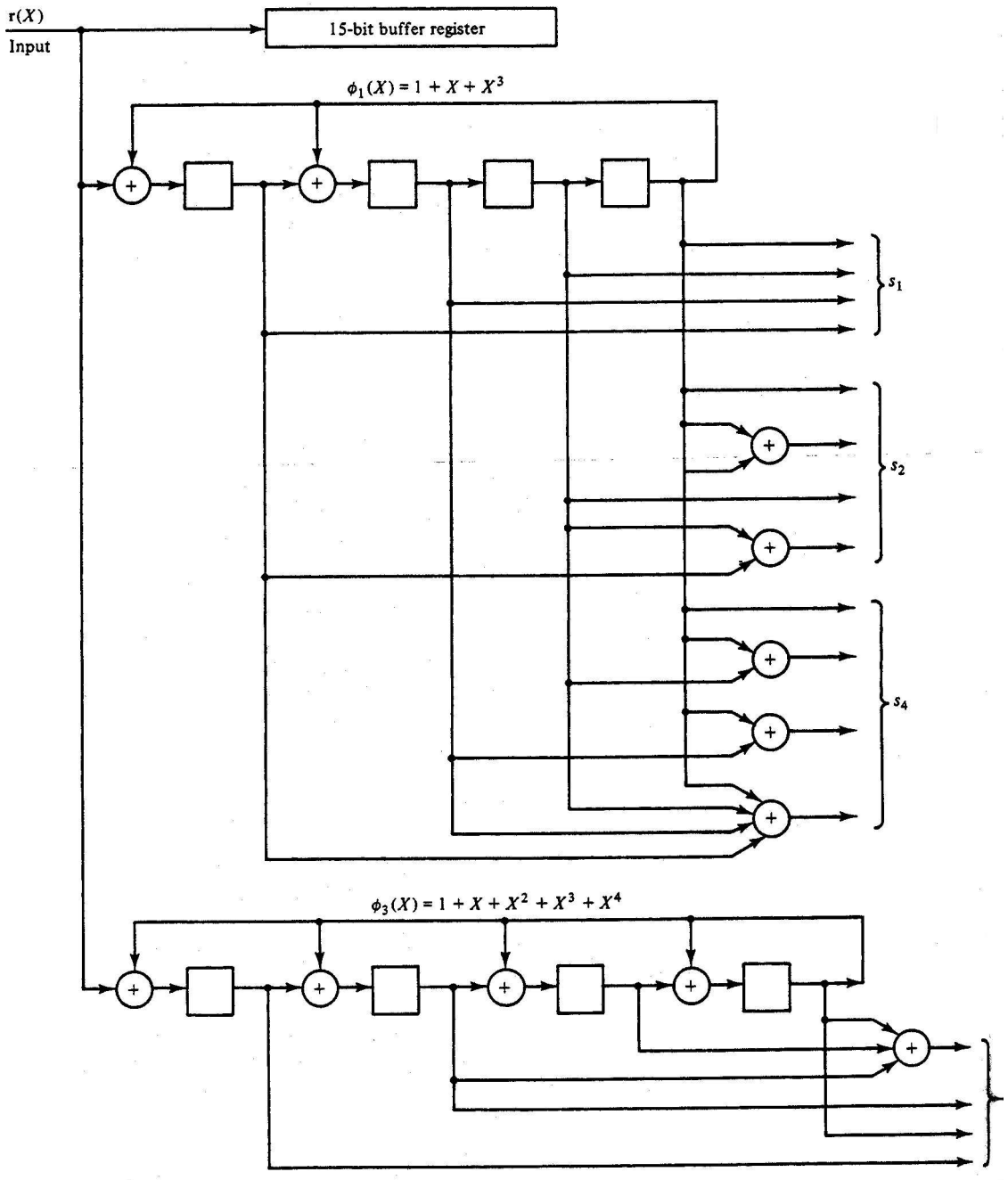
$$S_i = r(\alpha^i) = r_{n-1}(\alpha^i)^{n-1} + r_{n-2}(\alpha^i)^{n-2} + \ldots + r_1\alpha^i + r_0$$

$$= (\ldots((r_{n-1}\alpha^i + r_{n-2})\alpha^i + r_{n-3})\alpha^i + \ldots + r_1)\alpha^i + r_0$$

Αυτός υπολογισμός απαιτεί n – 1 προσθέσεις και n – 1 πολλαπλασιασμούς. Για δυαδικούς BCH κώδικες αποδεικνύεται ότι $S_{2i} = S_i^{\ 2}$. Βάσει αυτής της ισότητας τα 2t συστατικά του συνδρόμου υπολογίζονται με (n – 1)t προσθέσεις και nt πολλαπλασιασμούς.

Στην υλοποίηση με υλικό τα συστατικά του συνδρόμου μπορούν να υπολογιστούν με καταχωρητές ολίσθησης με ανάδραση όπως περιγράφηκε στην παράγραφο 6.3.Έχουμε τη δυνατότητα να χρησιμοποιήσουμε τα κυκλώματα των σχημάτων 6.7, 6.8 ή τον τύπο του κυκλώματος του σχήματος 6.10. Ο δεύτερος τύπος κυκλώματος είναι απλούστερος. Από τη έκφραση (6.3) βλέπουμε ότι το γενεσιουργό (generator) πολυώνυμο είναι το γινόμενο το πολύ t ελάχιστων πολυωνύμων. Επομένως το πολύ t καταχωρητές ολίσθησης με ανάδραση, κάθε ένας από τους οποίους το πολύ m καταστάσεις χρειάζονται για να διαμορφώσουν τα 2t συστατικά του συνδρόμου. Ο υπολογισμός γίνεται μόλις το λαμβανόμενο πολυώνυμο r(X) εισέλθει στον αποκωδικοποιητή. Για να ολοκληρωθούν οι υπολογισμοί απαιτούνται n κύκλοι ρολογιού. Ένα κύκλωμα υπολογισμού συνδρόμου για διόρθωση 2 λαθών (15,7) BCH κώδικα φαίνεται στο **σχήμα 6.11σελ168** όπου 2 καταχωρητές ολίσθησης με ανάδραση λειτουργούν με 4 καταστάσεις ο καθένας.

Το πλεονέκτημα του υπολογισμού με υλικό είναι η υπολογιστική ταχύτητα του συνδρόμου, ο υπολογισμός με προγραμματισμό είναι παρόλα αυτά πολύ πιο φθηνός.



Σχήμα 6.11 Κύκλωμα για τον υπολογισμό του συνδρόμου για ένα (15,7) BCH κώδικα που διορθώνει δύο λάθη.

Εύρεση του πολυώνυμου σ(X) θέσεων των λαθών

Σε αυτό το βήμα οι υπολογισμοί που απαιτούνται είναι λιγότεροι από t προσθέσεις και t πολλαπλασιασμούς για το κάθε $\sigma^{\mu}(X)$ και για το κάθε d_{μ}, και από τη στιγμή που έχουμε t από το καθένα συνολικά απαιτούνται $2t^2$ προσθέσεις και $2t^2$πολλαπλασιασμούς.Μια υλοποίηση βασισμένη καθαρά στο υλικό έχει το ίδιο σύνολο πράξεων και η ταχύτητα εξαρτάται από το πόσοι υπολογισμοί γίνονται παράλληλα. Ο τύπος κυκλώματος του σχήματος 6.2. χρησιμοποιείται για τις προσθέσεις και ο τύπος κυκλωμάτων των σχημάτων 6.5 και 6.6 για τους πολλαπλασιασμούς. Μια πολύ γρήγορη υλοποίηση υλικού όμως για τον προσδιορισμό του σ(X)θα ήταν πολύ πιθανόν πολύ ακριβή.

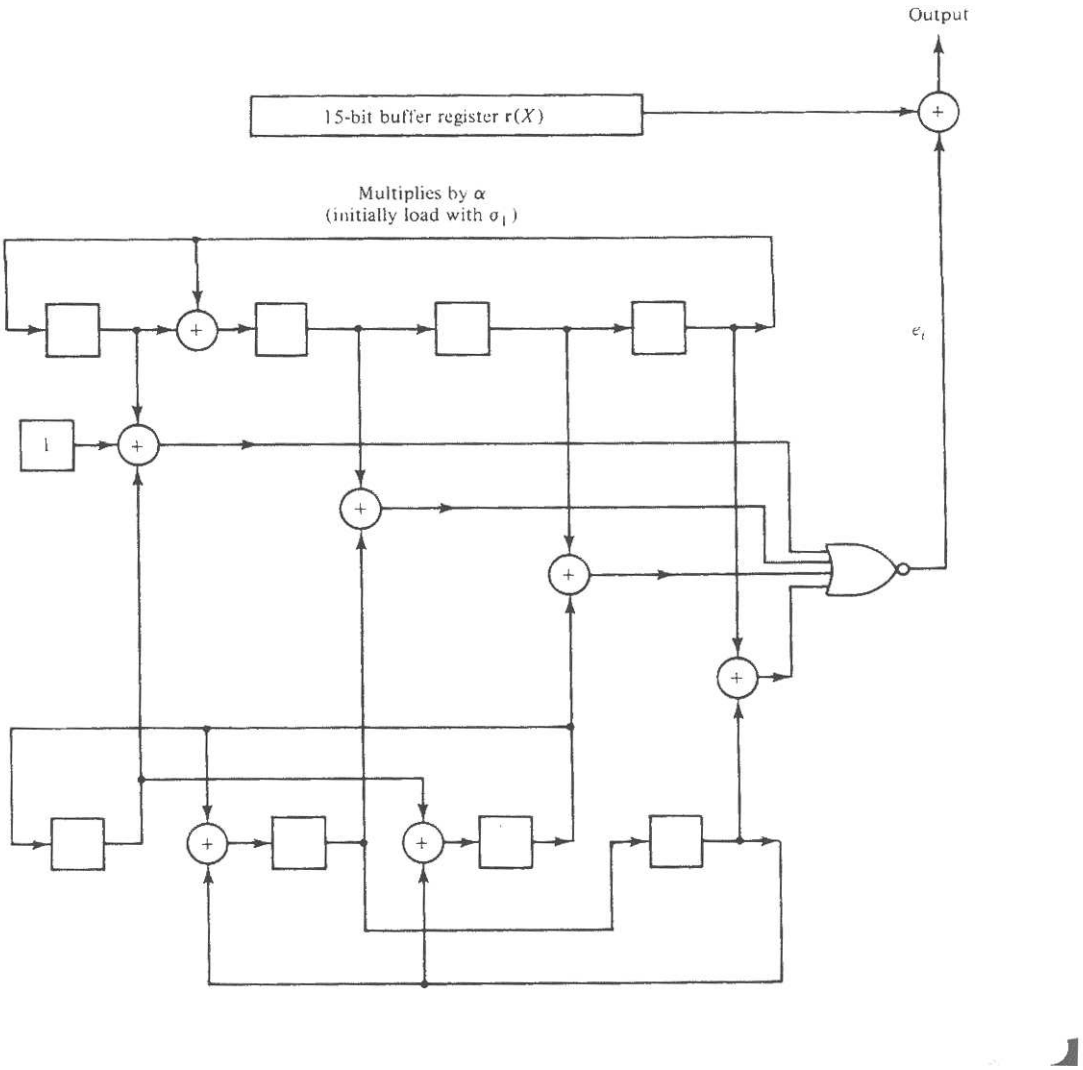
Υπολογισμός των αριθμών θέσης λαθών και διόρθωση των λαθών

Στην χειρότερη περίπτωση το βήμα αυτό απαιτεί την αντικατάσταση n στοιχείων του πεδίου στο πολυώνυμο σ(X) θέσης λαθών βαθμού t για να προσδιοριστούν οι ρίζες του.Στον προγραμματισμό αυτό απαιτεί nt προσθέσεις και nt πολλαπλασιασμούς. Στην περίπτωση υλικού μπορούμε να χρησιμοποιήσουμε το κύκλωμα του σχήματος 6.1 που διαθέτει t πολλαπλασιαστές για τον πολλαπλασιασμό με τα α^1, α^2,…, α^{2t} αντίστοιχα.Οι πολλαπλασιαστές μπορούν να είναι τύπου των σχημάτων 6.3 και 6.4.Αρχικά τα σ_1, σ_2,…, σ_t που βρέθηκαν στο δεύτερο βήμα φορτώνονται στους καταχωρητές των t πολλαπλασιαστών και στη συνέχεια ολισθαίνουν n φορές. Στο τέλος της *l* ολίσθησης οι t καταχωρητές θα περιέχουν $\sigma_1\alpha^l$, $\sigma_2\,\alpha^{2l}$,…, $\sigma_t\,\alpha^{tl}$. Τότε το άθροισμα

$$1+ \sigma_1\alpha^l+ \sigma_2\,\alpha^{2l}+\ldots+ \sigma_t\,\alpha^{tl}.$$

ελέγχεται. Αν το άθροισμα είναι μηδέν , το α^{n-l} είναι αριθμός θέσης λάθους διαφορετικά δεν θα είναι. Το άθροισμα αυτό επιτυγχάνεται με τη χρήση t modulo-2 αθροιστών m εισόδων. Μια πύλη OR m εισόδων θα ελέγξει αν το άθροισμα θα είναι μηδέν. Παίρνει n κύκλους ρολογιού για να ολοκληρωθεί αυτό το βήμα. Αν το μόνο που θέλουμε είναι η διόρθωση των ψηφίων μηνύματος,

μόνο κ κύκλοι ρολογιού χρειάζονται. Ένα κύκλωμα ανίχνευσης για διόρθωση δύο λαθών (15,7) BCH κώδικα φαίνεται στο **σχήμα 6.12**



Σχήμα 6.12 Κύκλωμα αναζήτησης για τον (15,7) BCH κώδικα που διορθώνει δύο λάθη

Για μεγάλα m και t το κόστος για τη δημιουργία t πολλαπλασιαστών που θα πολλαπλασιάζουν α^1, α^2,…, α^{2t} σε ένα κύκλο ρολογιού είναι πολύ μεγάλο. Για πιο οικονομικούς πολλαπλασιαστές αλλά και πιο αργούς χρησιμοποιούμε το κύκλωμα του σχήματος 6.5.Αρχικά το σ_i φορτώνεται στο καταχωρητή Β και το α^i αποθηκεύεται στον καταχωρητή C.Μετά από m κύκλους ρολογιού το γινόμενο $\sigma_i \alpha^i$ θα βρίσκεται στον καταχωρητή Α. Για να διαμορφωθεί το $\sigma_i \alpha^{2i}$, το $\sigma_i \alpha^i$ φορτώνεται στο καταχωρητή Β. Μετά από m κύκλους ρολογιού το $\sigma_i \alpha^{2i}$ θα βρίσκεται στον καταχωρητή Α.

Χρησιμοποιώντας αυτόν τον τύπο πολλαπλασιαστών θα χρειαστούν nm κύκλοι ρολογιού για να ολοκληρωθεί το τρίτο βήμα αποκωδικοποίησης ενός δυαδικού BCH κώδικα.

Τα βήματα 1 και 3 περιλαμβάνουν χονδρικά τον ίδιο αριθμό υπολογισμών. Από τη στιγμή που το n είναι γενικώς πολύ μεγαλύτερο από το t, 4nt είναι πολύ μεγαλύτερο από $4t^2$ και τα βήματα 1 και 3 περιλαμβάνουν πολλούς περισσότερους υπολογισμούς. Επομένως η υλοποίηση με υλικό αυτών των βημάτων είναι απαραίτητη μόνο αν απαιτείται υψηλή ταχύτητα αποκωδικοποίησης.

6.5 Κατανομή βάρους και ανίχνευση λαθών ενός δυαδικού BCH κώδικα

Η κατανομή βάρους έχει καθοριστεί πλήρως για BCH κώδικες που διορθώνουν δύο, τρία λάθη και για μερικούς χαμηλής συχνότητας βασικούς BCH κώδικες. Παρόλα αυτά για άλλους BCH κώδικες η κατανομή βάρους είναι ακόμα άγνωστη. Ο υπολογισμός της κατανομής βάρους για BCH κώδικες που διορθώνουν δύο, τρία λάθη επιτυγχάνεται με τον υπολογισμό της κατανομής βάρους του δυαδικού τους κώδικα και την εφαρμογή της στην ταυτότητα Mac Williams της (3.32). Η κατανομή βάρους του δυαδικού κώδικα ενός BCH κώδικα που διορθώνει δύο λάθη μήκους $2^m - 1$ δίνεται στους πίνακες 6.9 και 6.10. Η κατανομή βάρους του δυαδικού κώδικα ενός BCH κώδικα που διορθώνει τρία λάθη δίνεται στους πίνακες 6.11 και 6.12.

Πίνακας 6.9 Κατανομή βάρους ενός δυαδικού primitive BCH κώδικα μήκους $2^m - 1$ που διορθώνει δύο λάθη

Περιττός $m \geq 3$	
Βάρος i	Αριθμός διανυσμάτων με βάρος i, B_i
0	1
$2^m - 1 - 2^{(m+1)/2-1}$	$[2^{m-2} + 2^{(m-1)/2-1}](2^m - 1)$
$2^m - 1$	$[2^m - 2^{m-1} + 1](2^m - 1)$

Πίνακας 6.10 Κατανομή βάρους ενός δυαδικού primitive BCH κώδικα μήκους $2^m - 1$ που διορθώνει δύο λάθη

Ζυγός $m \geq 4$	
Βάρος i	Αριθμός διανυσμάτων με βάρος i, B_i
0	1
$2^{m-1} - 2^{(m+2)/2-1}$	$2^{(m-2)/2-1}[2^{(m-2)/2}+1](2^m-1)/3$
$2^{m-1} - 2^{m/2-1}$	$2^{(m+2)/2-1}[2^{m/2}+1](2^m-1)/3$
2^{m-1}	$(2^{m-2}+1)(2^m-1)$
$2^{m-1} + 2^{m/2-1}$	$2^{(m+2)/2-1}[2^{m/2}-1](2^m-1)/3$
$2^{m-1} + 2^{(m+2)/2-1}$	$2^{(m-2)/2-1}[2^{(m-2)/2}-1](2^m-1)/3$

Πίνακας 6.11 Κατανομή βάρους ενός δυαδικού primitive BCH κώδικα μήκους $2^m - 1$ που διορθώνει τρία λάθη

Περιττός $m \geq 5$	
Βάρος i	Αριθμός διανυσμάτων με βάρος i, B_i
0	1
$2^{m-1} - 2^{(m+1)/2}$	$2^{(m-5)/2}[2^{(m-3)/2}+1](2^{m-1}-1)(2^m-1)/3$
$2^{m-1} - 2^{(m-1)/2}$	$2^{(m-3)/2}[2^{(m-1)/2}+1](5\cdot 2^{m-1}+4)(2^m-1)/3$
2^{m-1}	$(9\cdot 2^{2m-4}+3\cdot 2^{m-3}+1)(2^m-1)$
$2^{m-1} + 2^{(m-1)/2}$	$2^{(m-3)/2}[2^{(m-1)/2}-1](5\cdot 2^{m-1}+4)(2^m-1)/3$
$2^{m-1} + 2^{(m+1)/2}$	$2^{(m-5)/2}[2^{(m-3)/2}-1](2^{m-1}-1)(2^m-1)/3$

Πίνακας 6.12 Κατανομή βάρους ενός δυαδικού primitive BCH κώδικα μήκους $2^m - 1$ που διορθώνει τρία λάθη

Ζυγός $m \geq 6$	
Βάρος i	Αριθμός διανυσμάτων με βάρος i, B_i
0	1
$2^{m-1} - 2^{(m+4)/2-1}$	$[2^{m-1} + 2^{(m+4)/2-1}] (2^m - 4)(2^m - 1)/960$
$2^{m-1} - 2^{(m+2)/2-1}$	$7[2^{m-1} + 2^{(m+2)/2-1}] 2^m (2^m - 1)/48$
$2^{m-1} - 2^{m/2-1}$	$2[2^{m-1} + 2^{m/2-1}] (3 \cdot 2^m + 8) (2^m - 1)/15$
2^{m-1}	$(29 \cdot 2^{2m} - 4 \cdot 2^m + 64) (2^m - 1)/64$
$2^{m-1} + 2^{m/2-1}$	$2[2^{m-1} - 2^{m/2-1}](3 \cdot 2^m + 8)(2^m - 1)/15$
$2^{m-1} + 2^{(m+2)/2-1}$	$7[2^{m-1} - 2^{(m+2)/2-1}] 2^m (2^m - 1) /48$
$2^{m-1} + 2^{(m+4)/2-1}$	$[2^{m-1} - 2^{(m+4)/2-1}] (2^m - 4)(2^m - 1)/960$

Αν ένας διπλής διόρθωσης λαθών ή τριπλής διόρθωσης λαθών BCH κώδικας χρησιμοποιηθεί για την ανίχνευση λαθών σε ένα δυαδικό συμμετρικό κανάλι (BSC) με πιθανότητα μετάδοσης ρ, η πιθανότητα ενός μη ανιχνεύσιμου λάθους υπολογίζεται από την

$$P_u(E) = 2^{-(n-k)} B(1-2p) - (1-p)^n$$

Έχει αποδειχτεί ότι η πιθανότητα ενός μη ανιχνεύσιμου λάθους $P_u(E)$ ενός BCH κώδικα που διορθώνει δύο λάθη μήκους $2^m - 1$είναι φραγμένη άνω από 2^{-2m} για $\rho \leq 1/2$ όπου 2m είναι ο αριθμός των ψηφίων ελέγχου ισοτιμίας του κώδικα. Είναι άγνωστο για το αν η πιθανότητα ενός μη ανιχνεύσιμου λάθους $P_u(E)$ ενός BCH κώδικα που διορθώνει τρία λάθη μήκους $2^m - 1$ικανοποιεί το άνω φράγμα 2^{-3m} όπου 3m είναι ο αριθμός των ψηφίων ελέγχου ισοτιμίας του κώδικα.

Στο σημείο αυτό μελετάμε την απόδοση ενός βασικού BCH κώδικα που διορθώνει t λάθη όταν χρησιμοποιείται για ανίχνευση λαθών σε ένα δυαδικό συμμετρικό κανάλι με πιθανότητα μετάδοσης ρ. Έχει αποδειχτεί ότι ένας βασικός BCH κώδικας που διορθώνει t λάθη μήκους $2^m - 1$, εάν ο αριθμός των ψηφίων ελέγχου ισοτιμίας του ισούται με mt και m είναι μεγαλύτερο από μια

συγκεκριμένη σταθερά $m_0(t)$, ο αριθμός των κωδικών διανυσμάτων βάρους i ικανοποιεί τις παρακάτω ισότητες.

$A_i = 0$ για $0 \le i \le 2t$

$$A_i = (1+ \lambda_o n^{-1/10}) \binom{n}{i} 2^{-(n-k)} \text{ για } i > 2t \qquad (6.36)$$

Όπου $n=2^m - 1$ και λ_o είναι άνω φραγμένη από ένα σταθερό αριθμό.

Από την παραπάνω ισότητα και την (3.19) εξάγεται η επόμενη έκφραση για την πιθανότητα ενός μη ανιχνεύσιμου λάθους.

$$P_u(E) = (1+ \lambda_o n^{-1/10}) \binom{n}{i} 2^{-(n-k)} \sum_{i=2t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i} \qquad (6.37)$$

Έστω ότι $\varepsilon = (2t+1)/n$ Τότε το άθροισμα της (6.37) μπορεί να έχει ως άνω όριο το:

$$\sum_{i=2t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i} \le 2^{-nE(\varepsilon,\rho)} \qquad (6.38)$$

δεδομένου $\rho < \varepsilon$

$E(\varepsilon,\rho) = H(\rho)+(\varepsilon - \rho) H'(\rho) - H(\varepsilon)$

$H(x)= - x\log_2 x - (1 - x)\log_2(1 - x)$

Και $H'(x) = \log_2 \frac{1-x}{x}$

Το $E(\varepsilon,\rho)$ είναι θετικό για κάθε $\rho < \varepsilon$ Από τις (6.37) και (6.38) προκύπτει το επόμενο άνω φράγμα της $P_u(E)$ για $\rho < \varepsilon$.

$$P_u(E) \le (1+ \lambda_o n^{-1/10}) 2^{-nE(\varepsilon,\rho)} 2^{-(n-k)} \qquad (6.39)$$

Για $\rho < \varepsilon$ και ικανοποιητικά μεγάλο n, το $P_u(E)$ μπορεί να γίνει πολύ μικρό. Για $\rho \ge \varepsilon$ το φράγμα για το $P_u(E)$ θα είναι

$P_u(E) \leq (1+ \lambda_o n^{-1/10}) 2^{-(n-k)} \sum_{i=2t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$

Αν $\sum_{i=2t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i} = 1$ το όριο θα γίνει

$P_u(E) \leq (1+ \lambda_o n^{-1/10}) 2^{-(n-k)}$

Για $\rho \geq \varepsilon$, $P_u(E)$ θα μειωθεί εκθετικά με τον αριθμό των ψηφίων ελέγχου ισοτιμίας, n – k.Αν χρησιμοποιήσουμε ικανοποιητικά μεγάλο αριθμό ψηφίων ελέγχου ισοτιμίας, η πιθανότητα ενός μη ανιχνεύσιμου λάθους $P_u(E)$ θα γίνει πολύ μικρή. Άρα περιληπτικά τα αποτελέσματα όλων των παραπάνω είναι :

Για ένα BCH κώδικα που διορθώνει t λάθη μήκους $2^m - 1$ με αριθμό ψηφίων ελέγχου ισοτιμίας,

$n - k = mt$ και $m \geq m_0(t)$, η πιθανότητα ενός μη ανιχνεύσιμου λάθους $P_u(E)$ σε ένα δυαδικό συμμετρικό κανάλι με πιθανότητα μετάδοσης p ικανοποιεί τα ακόλουθα όρια:

$P_u(E) \leq (1+ \lambda_o n^{-1/10}) 2^{-n(1-R)}$ για $\rho \geq \varepsilon$

$P_u(E) \leq (1+ \lambda_o n^{-1/10}) 2^{-n(1-R+ E(\varepsilon,\rho))}$ για $\rho < \varepsilon$

Όπου $\varepsilon = (2t+1)/n$, $R= k/n$ και λ_o είναι μία σταθερά.

Επομένως η παραπάνω ανάλυση υποδεικνύει ότι οι βασικοί BCH κώδικες είναι πολύ αποτελεσματικοί στην ανίχνευση λαθών στα δυαδικά συμμετρικά κανάλια. Παρόλο που δεν έχει αποδειχθεί θεωρείται ότι ισχύει η πιθανότητα ενός μη ανιχνεύσιμου λάθους για κάθε βασικό BCH κώδικα ικανοποιεί το άνω όριο $2^{-(n-k)}$.

ΚΕΦΑΛΑΙΟ 7 –

Στα πλαίσια της παρούσας εργασίας υλοποιήθηκε ένα σύνολο βιβλιοθηκών για την επικοινωνία δύο απομακρυσμένων υπολογιστών μέσω του πρωτοκόλλου RS232. Οι μεταδιδόμενες πληροφορίες κωδικοποιούνται με BCH κώδικα.

Το αποτέλεσμα της υλοποίησης είναι η ύπαρξη ενός εύχρηστου και απλού μηχανισμού για την αξιόπιστη επικοινωνία ακόμα και σε συνθήκες υψηλού θορύβου στο κανάλι επικοινωνίας. Η υλοποίηση είναι κατάλληλη για ασύρματη επικοινωνία χρησιμοποιώντας κατάλληλους πομποδέκτες στη σειριακή έξοδο του υπολογιστή.

Σε αυτό το κεφάλαιο θα παρουσιαστούν τα χαρακτηριστικά της υλοποίησης καθώς και μία απλή εφαρμογή επικοινωνίας που αναπτύχθηκε με σκοπό την επίδειξη της λειτουργικότητας.

7.1 Προδιαγραφές υλοποίησης

Η υλοποίηση επιχειρεί να καλύψει τις ανάγκες εφαρμογών απομακρυσμένης παρακολούθησης και αυτόματου ελέγχου. Τα χαρακτηριστικά τέτοιων εφαρμογών περιλαμβάνουν:

- Χαμηλής υπολογιστικής ισχύος υπολογιστές τουλάχιστον στο ένα άκρο επικοινωνίας. Οι υπολογιστές που είναι υπεύθυνοι για τον έλεγχο των συγκεκριμένων συσκευών μπορεί να διαθέτουν πολύ περιορισμένη υπολογιστική ισχύ και να μη διαθέτουν τους απαραίτητους πόρους για τη λειτουργία σύγχρονων λειτουργικών συστημάτων.

- Εξασφάλιση επικοινωνίας σε μεγάλες αποστάσεις. Η επικοινωνία πρέπει να γίνεται ακόμα και από μεγάλες αποστάσεις οι οποίες, λόγω της ιδιομορφίας του εδάφους ή της έλλειψης σχετικής υποδομής, δεν μπορεί να καλυφθούν με ενσύρματη επικοινωνία. Αυτό συνοδεύεται συνήθως από ασύρματες συνδέσεις με χαμηλής αξιοπιστίας κανάλια επικοινωνίας.

- Τα δεδομένα που χρειάζεται να μεταδίδονται είναι μικρά σε μέγεθος και η συχνότητα επικοινωνίας μικρή. Από την άλλη απαιτείται μεγάλη αξιοπιστία στη μετάδοση ειδικά εάν οι υπό έλεγχο συσκευές είναι υπεύθυνες για κρίσιμες λειτουργίες.

Τα παραπάνω χαρακτηριστικά οδήγησαν στις ακόλουθες προδιαγραφές για την υλοποίηση του μηχανισμού επικοινωνίας:

Η επικοινωνία γίνεται δια μέσω του RS232 πρωτοκόλλου το οποίο είναι διαθέσιμο για τα περισσότερα υπολογιστικά συστήματα. Στις σχετικές θύρες μπορεί εύκολα να προσαρμοστούν ιδιαίτερα φθηνές συσκευές ασύρματης μετάδοσης των σημάτων επικοινωνίας.

Η προς μετάδοση πληροφορία κωδικοποιείται με BCH (31,21,2) κώδικα ο οποίος εξασφαλίζει επαρκές μέγεθος πληροφορίας μετάδοσης (21 bits) για τις συγκεκριμένες εφαρμογές και ανιχνεύει και διορθώνει μέχρι και 2 bit σε σφάλματα επικοινωνίας χωρίς να υπάρχει ανάγκη επαναμετάδοσης. Η επιλογή του BCH κώδικα έναντι των άλλων προτεινόμενων κωδίκων στη συγκεκριμένη υλοποίηση βασίστηκε στο γεγονός ότι οι βασικοί BCH κώδικες είναι πολύ αποτελεσματικοί στην ανίχνευση λαθών στα δυαδικά συμμετρικά κανάλια και ως κυκλικοί κώδικες διαθέτουν αρκετές αλγεβρικές και γεωμετρικές ιδιότητες που απλοποιούν στο μέγιστο βαθμό τη διαδικασία αποκωδικοποίησης.

Η υλοποίηση έγινε για απλό λειτουργικό σύστημα DOS το οποίο έχει ιδιαίτερα μικρές απαιτήσεις σε υπολογιστικούς πόρους.

7.2 Κωδικοποίηση

Η υλοποίηση της κωδικοποίησης με BCH (31, 21, 2) κώδικα βασίστηκε στο σχετικό πηγαίο κώδικα του Morelos-Zaragoza. Για τον προσδιορισμό του πολυωνύμου θέσεων λαθών (error locator polynomial) δε χρησιμοποιείται ο επαναληπτικός αλγόριθμος αλλά λύνονται οι 2 εξισώσεις για την εύρεση των θέσεων των λαθών.

Στον παραπάνω πηγαίο κώδικα έγιναν οι κατάλληλες τροποποιήσεις ώστε να δημιουργηθεί ένα C include αρχείο το οποίο μπορεί στη συνέχεια να χρησιμοποιηθεί σε άλλες εφαρμογές. Προστέθηκαν επίσης συναρτήσεις για κωδικοποίηση και αποκωδικοποίηση έχοντας τα δεδομένα σε συνεχόμενα διανύσματα χαρακτήρων (αντί της χρήσης ενός ακεραίου για κάθε bit δεδομένων), για να είναι εύκολη η μετάδοση από τη σειριακή θύρα.

Οι σχετικές συναρτήσεις έχουν ως είσοδο ένα διάνυσμα από 3 bytes για την κωδικοποίηση και από 4 bytes για την αποκωδικοποίηση. Από τα 3 byte τα αρχικά 3 bit παραλείπονται για να

δημιουργηθεί η ακολουθία των 21 bit του BCH κώδικα. Από τα 4 byte το αρχικό bit παραλείπεται για να δημιουργηθεί η ακολουθία των bit της κωδικοποιημένης BCH λέξης.

7.3 Σειριακή επικοινωνία

Η σειριακή επικοινωνία βασίστηκε στο σχετικό πηγαίο κώδικα του C. Karcher. Δε χρησιμοποιήθηκαν οι λειτουργίες για έλεγχο ροής ενώ γίνεται εκμετάλλευση μόνο των σημάτων για αποστολή, λήψη και γείωση καθώς δεν είναι διαθέσιμα τα επιπλέον σήματα στην ασύρματη επικοινωνία.

Ως βασική λέξη επικοινωνίας χρησιμοποιείται μία ακολουθία από 4 byte ώστε να καλύπτεται η λέξη του BCH κώδικα. Η κωδική αυτή λέξη μεταδίδεται στον παραλήπτη. Εκεί αποκωδικοποιείται, αφού πρώτα ελεγχθεί για λάθη μετάδοσης και διορθωθεί και παραδίδεται στον παραλήπτη. Ο παραλήπτης με τη σειρά του αποστέλλει μία επιβεβαίωση λήψης στον αποστολέα (ACK) ότι παρέλαβε την πληροφορία σωστά. Αν ο αριθμός λαθών ξεπεράσει το όριο αυτών που ο BCH κώδικας μπορεί να διορθώσει τότε ο παραλήπτης στέλνει αρνητική επιβεβαίωση (NACK) στον αποστολέα και ζητά την επαναμετάδοση της πληροφορίας.

Τόσο κατά την αποστολή όσο και κατά τη λήψη δεδομένων υπάρχει ένα άνω χρονικό όριο που όταν ξεπεραστεί χωρίς να έχει γίνει η αποστολή ή η λήψη της κωδικής λέξης, οι σχετικές συναρτήσεις επιστρέφουν κατάλληλο μήνυμα λάθους (timeout).

7.4 Εφαρμογή επίδειξης

Για το σκοπό της επίδειξης της λειτουργίας των βιβλιοθηκών που υλοποιήθηκαν, αναπτύχθηκε μία απλή εφαρμογή απομακρυσμένης συνομιλίας κειμένου (text chat). Το κάθε άκρο είναι υπεύθυνο αφ’ ενός να αποστέλλει όσους χαρακτήρες πληκτρολογούνται από το χρήστη στο άλλο άκρο, και αφ’ ετέρου να λαμβάνει όσους χαρακτήρες στέλνονται από το άλλο άκρο και να τους εμφανίζει στην οθόνη.

Το κάθε μήνυμα που αποστέλλεται έχει ως πληροφορία ένα byte για τον κάθε χαρακτήρα που πληκτρολογείται. Το byte πληροφορίας κωδικοποιείται με τη χρήση των βιβλιοθηκών και αποστέλλεται στο άλλο άκρο όπου ελέγχεται για τυχών λάθη μετάδοσης και διορθώνεται ή αποβάλλεται εάν δεν μπορεί να διορθωθεί.. Στην περίπτωση αποβολής ζητείται η επαναμετάδοση.

8 Βιβλιογραφία

Walrand Jean: Communication Networks, Irwin, Boston 1991.

Pless Vera : Indroduction to the theory of error correcting codes 2nd ed., University of Illinois 1989.

Shu Lin /Daniel j. Costello,Jr :Error control Coding: Fundamentals and Applications, University of Hawaii Texas A&M University/Illinois Institute of Technology 1983.

Mano: Ψηφιακή σχεδίαση 2η έκδοση Prentice- Hall- Παπασωτηρίου, Αθήνα 1992.

Simon Haykin : Συστήματα επικοινωνίας, Ε.Δ.Συκάς, Μ.Ε.Θεολόγου Ε.Μ.Π, Παπασωτηρίου, Αθήνα 1995.

Andy Bateman:Ψηφιακές Επικοινωνίες Σχεδίαση συστημάτων στην πράξη, Τζιόλα Θεσσαλονίκη 1999.

Taub/Schilling:Τηλεπικοινωνιακά Συστήματα 2η έκδοση, Τζιόλα Θεσσαλονίκη 1998.

Morelos-Zaragoza, BCH(31, 21, 2), http://www.eccpage.com

C. Karcher, RS232 Set of general purpose functions providing fully buffered interrupt driven serial I/0