ΚΩΔΙΚΟΠΟΙΗΣΗ ΚΑΝΑΛΙΟΥ

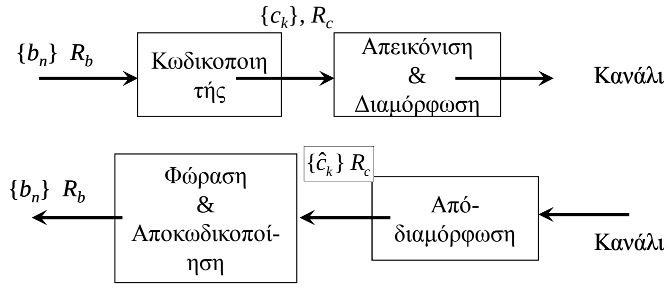
(CHANNEL CODING**)**

Ο όρος Κωδικοποίηση Καναλιού αναφέρεται κυρίως σε κώδικες *Πρόσθιας Διόρθωσης Σφαλμάτων* (*Forward Error Correction*) και την *Σύμπλεξη Δυαδικών Ψηφίων* (*Bit Interleaving*) κατά τη διαβίβαση δεδομένων μέσα από κανάλια, ή κατά την φύλλαξη και ανάκτηση δυαδικών δεδομένων σε μέσα αποθήκευσης.

Οι κώδικες καναλιού χρησιμοποιούνται για την προστασία των δυαδικών δεδομένων όχι μόνο από τον θόρυβο, αλλά και από φαινόμενα διαλείψεων ή μεταβολών του καναλιού.

Οι κώδικες καναλιού εισάγουν πλεονάζοντα κωδικά bits με δομημένο τρόπο, ώστε να αντιμετωπίζονται αποτελεσματικά οι παραμορφώσεις και ο θόρυβος του καναλιού.

<u>Γενικό Διάγραμμα Συστήματος Ψηφιακών Επικοινωνιών με Βαθμίδες Κωδικοποίησης & Αποκωδικοποίησης.</u>



Λόγω της εισαγωγής του πλεονασμού των δυαδικών bits ισχύει:

$$R_c > R_b$$

Η χρήση δηλαδή του κώδικα καναλιού έχει πάντα ως αποτέλεσμα την αύξηση του ρυθμού διαβίβασης δυαδικών ψηφίων και συνήθως απαιτεί αύξηση του απαιτούμενου εύρους ζώνης.

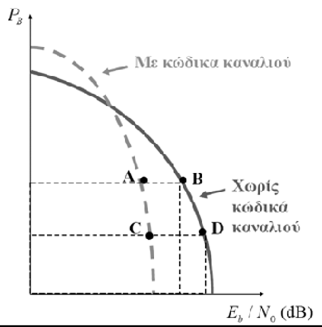
Το πηλίκο R_b/R_c καλείται *Ρυθμός* του Κώδικα (*Coding Rate*) και έχει τιμή μικρότερη από τη μονάδα.

Ως αντάλλαγμα στο αυξημένο εύρος ζώνης και στην πολυπλοκότητα του κώδικα, επιδιώκουμε να επιτύχουμε τη διαβίβαση της ακολουθίας πληροφορίας $\{b_n\}$ με ελαττωμένη ενέργεια ανά bit, E_b διατηρώντας την τιμή στόχου της πιθανότητα σφάλματος P_b.

Τη δαπανούμενη ενέργεια ανά διαβιβαζόμενο bit της ακολουθίας πληροφορίας $\{b_n\}$ με κωδικοποίηση συμβολίζουμε με $E_{\text{b-coded}}$ και χωρίς κωδικοποίηση τη συμβολίζουμε με $E_{\text{b-uncoded}}$.

Το πηλίκο $E_{\text{b-uncoded}}/E_{\text{b-coded}}$ καλείται *Κωδικό Κέρδος* (*Coding Gain*).

Για έναν κώδικα με σωστή λειτουργία, πρέπει το κέρδος του κώδικα να είναι μεγαλύτερο από τη μονάδα, ή να έχει θετική τιμή σε dB.



<u>Παράδειγμα Κώδικα</u>

Data Words		Code Words				
0	0	0	0	0	0	0
0	1	0	1	0	1	1
1	0	1	0	1	0	1
1	1	1	1	1	1	0

Ρυθμός Κώδικα ρ=2/5

Παρατηρείστε ότι κάθε ζευγάρι κωδικών λέξεων διαφέρουν τουλάχιστον σε τρία bits

Απόσταση Hamming, $d_H(\mathbf{c}_1,\mathbf{c}_2)$, μεταξύ δύο κωδικών λέξεων $\mathbf{c}_1,\mathbf{c}_2$ καλείται το πλήθος των διαφορετικών bits μεταξύ αυτών.

Ελάχιστη *Απόσταση Hamming*, d^H_{min}, του κώδικα καλείται η ελάχιστη τιμή των αποστάσεων Hamming για όλα τα δυνατά ζεύγη των κωδικών λέξεων.

Στο κώδικα του Παραδείγματος $d^H_{min}=3$

Data Words		Code Words				
0	0	0	0	0	0	0
0	1	0	1	0	1	1
1	0	1	0	1	0	1
1	1	1	1	1	1	0

Αν, λοιπόν, κατά τη διαβίβαση μιας κωδικής λέξης συμβεί ένα μόνο σφάλμα, η λέξη λήψης που θα προκύψει θα διαφέρει από τη σωστή κωδική λέξη σε 1 bits και από όλες τις υπόλοιπες τουλάχιστον σε 2 θέσεις. Εύκολα λοιπό εντοπίζεται η σωστή κωδική λέξη και επομένως η αντίστοιχη λέξη δεδομένων.

Από την άλλη πλευρά είναι ενδιαφέρον να παρατηρήσετε ότι αν συμβούν δύο λάθη η λέξη λήψης δεν έχει πλέον ελάχιστη απόσταση από τη σωστή κωδική λέξη και δεν μπορεί να γίνει διόρθωση! Ωστόσο διαφέρει από όλες τις κωδικές λέξεις και ο δέκτης μπορεί να διαπιστώσει ότι συνέβη λάθος κατά τη διαβίβαση.

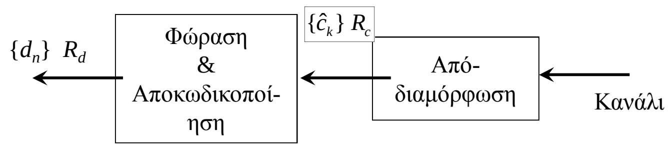
Data Words		Code Words				
0	0	0	0	0	0	0
0	1	0	1	0	1	1
1	0	1	0	1	0	1
1	1	1	1	1	1	0

Αν τα λάθη στη λέξη λήψης είναι περισσότερα από δύο υπάρχει πιθανότητα αυτή να γίνει ίδια με μια από τις άλλες κωδικές λέξεις οπότε ο δέκτης διαπιστώνει σωστή λήψη.

Ο δέκτης εργάζεται πάντα με έναν από τους δύο πιο κάτω τρόπους:

1. <u>Διόρθωση</u> Συγκρίνει τη λέξη λήψης με τις κωδικές και την αντικαθιστά την με την κωδική που απέχει λιγότερο
2. <u>Ανίχνευση Λάθους και Αίτηση Επανεκπομπής.</u> Αν η λέξη λήψης δεν συμπίπτει με καμία από τις κωδικές ζητά από τον πομπό την επανεκπομπή.

Hard και Soft Αποκωδικοποίηση



Θυμηθείτε το τμήμα Φώρασης και Αποκωδικοποίησης του δέκτη. Αυτό μπορεί να σχεδιαστεί με δύο διαφορετικούς τρόπους. Στην πρώτη τεχνική γίνεται πρώτα η φώραση και στη συνέχεια η αποκωδικοποίηση. Η τεχνική αυτή είναι γνωστή ως Hard Αποκωδικοποίηση.

Στη δεύτερη τεχνική η φώραση και η αποκωδικοποίηση γίνεται συγχρόνως. Για παράδειγμα αν η διαμόρφωση είναι με αντίποδα σύμβολα και χρησιμοποιείται ο κώδικας του παραδείγματος, λαμβάνεται διάνυσμα με πέντε δείγματα (r_1, r_2, r_3, r_4, r_5) και στη συνέχεια συγκρίνεται με τον αστερισμό των 4 κωδικών λέξεων:

Data Words		Αστερισμός Κωδικών Λέξεων				
0	0	-A	-A	-A	-A	-A
0	1	-A	A	-A	A	A
1	0	A	-A	A	-A	A
1	1	A	A	A	A	-A

Το πλέον πιθανό αποτέλεσμα είναι η κωδική λέξη του αστερισμού που απέχει ελάχιστα από το διλανυσμα λήψης $\mathbf{r}$=(r_1, r_2, r_3, r_4, r_5)

Η τελευταία τεχνική αποκωδικοποίησης είναι πιο πολύπλοκη, αλλά είναι η πιο αποτελεσματική, δηλαδή συμβαίνει σφάλμα πιο σπάνια από ό,τι στη hard αποκωδικοποίηση. Η τεχνική αυτή καλείται *Soft Αποκωδικοποίηση*.

Bit Error Probability για Κωδικοποίηση με hard Αποκωδικοποίηση.

Αν E_C είναι η δαπανούμενη ενέργεια ανά κωδικό bit για τη διαβίβαση των κωδικών bits, και αν δεχθούμε ότι η διαμόρφωση χρησιμοποιεί αντίποδα σύμβολα, η πιθανότητα εσφαλμένης λήψης ενός κωδικού bit, P_C, δίνεται από τη σχέση.

$$P_C = Q\left(\sqrt{\frac{2E_C}{N_0}}\right)$$

Για να συμβεί σφάλμα στην αποκωδικοποίηση πρέπει να συμβούν περισσότερα από ένα σφάλματα στην διαβιβαζόμενη κωδική λέξη. Επομένως η πιθανότητα σφάλματος P_e ισούται με:

$$P_e = \Pr\left[\text{να συμβούν 2, 3, 4, ή 5 σφάλματα στην κωδική λέξη}\right]$$

ή ισοδύναμα

$$P_e = 1 - \Pr\left[\text{να συμβούν 0, ή 1 σφάλματα στη κωδική λέξη}\right]$$

ή ισοδύναμα

$$P_e = 1 - \Pr\left[\text{ 0 σφάλματα}\right] - \Pr\left[\text{ 1 σφάλματα}\right]$$

και τελικά

$$P_e = 1 - \left(1 - P_C\right)^5 - 5P_C\left(1 - P_C\right)^4$$

Επειδή σε κάθε λάθος κωδικής λέξης κατά μέσον όρο τα μισά bit πληροφορίας είναι λάθος μπορούμε να δεχθούμε ότι ισχύει:

$$P_b = P_e / 2$$

Επειδή για τη διαβίβαση των 5 κωδικών bits πρέπει να χρησιμοποιηθεί συνολικά ενέργεια ίση με την ενέργεια διαβίβασης 2 bit πληροφορίας στο ακωδικοποίητο σήμα, ισχύει $E_C=(2/5)E_b$:

$$P_C = Q\left(\sqrt{\frac{4E_b}{5N_0}}\right)$$

και

$$P_e = 1 - \left(1 - Q\left(\sqrt{\frac{4E_b}{5N_0}}\right)\right)^5 - 5Q\left(\sqrt{\frac{4E_b}{5N_0}}\right)\left(1 - Q\left(\sqrt{\frac{4E_b}{5N_0}}\right)\right)^4$$

και τέλος

$$P_b = \left[1 - \left(1 - Q\left(\sqrt{\frac{4E_b}{5N_0}}\right)\right)^5 - 5Q\left(\sqrt{\frac{4E_b}{5N_0}}\right)\left(1 - Q\left(\sqrt{\frac{4E_b}{5N_0}}\right)\right)^4 \Big/ 2\right]$$

Soft Αποκωδικοποίηση Κέρδος Κωδικοποίησης.

Το Κέρδος Κωδικοποίησης για μεγάλες τιμές του E_b/N_0 μπορεί να υπολογιστεί με τη βοήθεια της ελάχιστης απόστασης. Έτσι για τον κώδικα (5,2) του παραδείγματος με d_{Hmin}=3

Ο αστερισμός των κωδικών λέξεων εκτείνεται στο χώρο των 5 διαστάσεων. Κάθε σύμβολο-κωδική λέξη έχει συντεταγμένες sqrt($\mathcal{E}_c$) ή -sqrt($\mathcal{E}_c$) (για κωδικά bits 1 και 0 αντίστοιχα). Από την παρατήρηση αυτή προκύπτει:

$$d_{\min}^2 = d_{H-\min} \times 4 \times \mathcal{E}_c = 12\mathcal{E}_c$$

Απόδειξη στην επόμενη διαφάνεια

Επειδή με τα 5 κωδικά bits διαβιβάζονται 2 bits πληροφορίας, ισχύει:

$\varepsilon_b = 5\varepsilon_c/2$ οπότε $\dfrac{d_{\min}^2}{\mathcal{E}_b} = 24/5 = 4.8$

Θυμηθείτε ότι στα αντίποδα σήματα ισχύει:

$$\left(\frac{d_{\min}^2}{\mathcal{E}_b}\right)_{uncoded} = 4$$

Το κέρδος του κώδικα επομένως είναι:

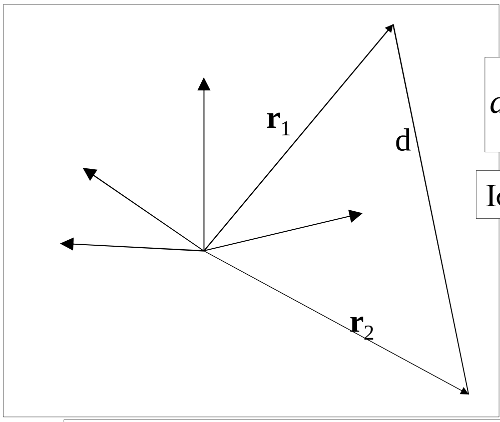
$$G = \left(\frac{d_{\min}^2}{\mathcal{E}_b}\right)_{coded} \Big/ \left(\frac{d_{\min}^2}{\mathcal{E}_b}\right)_{uncoded} = 4.8/4 = 1.2 = 0.8\text{dB}$$

Υπολογισμός της Ελάχιστης Απόστασης d_{min} για τον (5,2) Κώδικα

Ας θεωρήσουμε τον αστερισμό των συμβόλων κωδικών λέξεων που χρησιμοποιούμε στη soft αποκωδικοποίηση του παραπλεύρως Σχήματος.

Data Words		Αστερισμός Κωδικών Λέξεων				
0	0	$-\sqrt{E_c}$	$-\sqrt{E_c}$	$-\sqrt{E_c}$	$-\sqrt{E_c}$	$-\sqrt{E_c}$
0	1	$-\sqrt{E_c}$	$\sqrt{E_c}$	$-\sqrt{E_c}$	$\sqrt{E_c}$	$\sqrt{E_c}$
1	0	$\sqrt{E_c}$	$-\sqrt{E_c}$	$\sqrt{E_c}$	$-\sqrt{E_c}$	$\sqrt{E_c}$
1	1	$\sqrt{E_c}$	$\sqrt{E_c}$	$\sqrt{E_c}$	$\sqrt{E_c}$	$-\sqrt{E_c}$

Αυτά τα σύμβολα κωδικές λέξεις ανήκουν σε ένα χώρο πέντε διαστάσεων. Ας προσπαθήσουμε να υπολογίσουμε την απόσταση μεταξύ δύο οποιοδήποτε συμβόλων $\mathbf{r}_1$ και $\mathbf{r}_2$, $d(\mathbf{r}_1, \mathbf{r}_2)$



$$d(\mathbf{r}_1,\mathbf{r}_2)^2 = \|\mathbf{r}_1 - \mathbf{r}_2\|^2 = \sum_{i=1}^{5}(r_{1i} - r_{2i})^2$$

Ισχύει

$$r_{ji} = c_{ji}\sqrt{\varepsilon_c} \quad j = 1,2 \quad i = 1,2,\dots,5$$

$$c_{ji} = 1 \quad ή \quad c_{ji} = -1$$

Επομένως:

$$d(\mathbf{r}_1,\mathbf{r}_2)^2 = \|\mathbf{r}_1 - \mathbf{r}_2\|^2 = \varepsilon_c \sum_{i=1}^{5}(c_{1i} - c_{2i})^2 = \varepsilon_c 4 d_H(\mathbf{c}_1,\mathbf{c}_2)$$

$$d_H(\mathbf{c}_1,\mathbf{c}_2) \geq d_{\min}^H \Rightarrow d_{\min}^2 = 4 d_{\min}^H \varepsilon_c$$

Γενικά για Κώδικα (n,k) με ελάχιστη απόσταση Hamming: d^H_{min}

$$d^2_{min} = d^H_{min} \times 4 \times \mathcal{E}_c$$

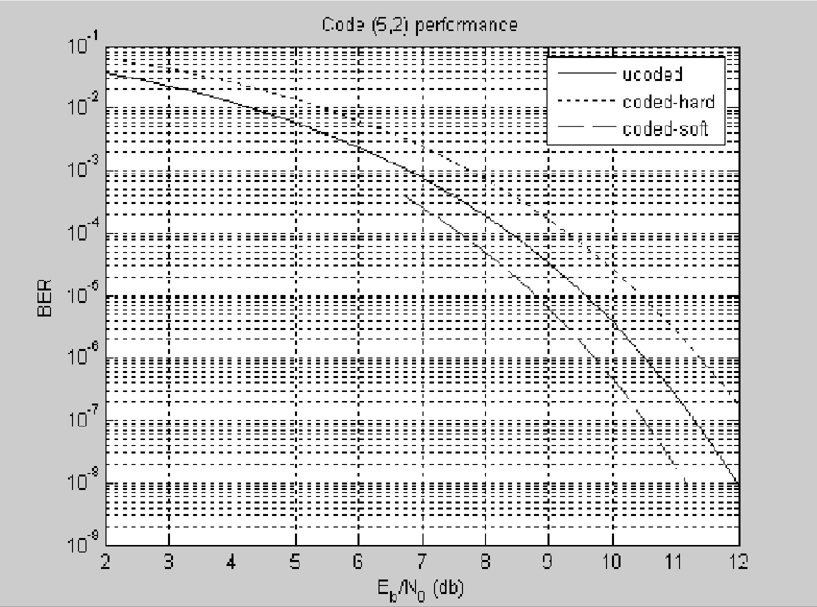
$$\mathcal{E}_b = n\mathcal{E}_c/k$$

$$\frac{d^2_{min}}{\mathcal{E}_b} = \frac{k}{n} d_{H\,min} \times 4$$

$$\left(\frac{d^2_{min}}{\mathcal{E}_b}\right)_{coded} = \frac{k}{n} d^H_{min} \times \left(\frac{d^2_{min}}{\mathcal{E}_b}\right)_{uncoded}$$

$$\frac{k}{n} d^H_{min} = G_{code}\left(\text{Κέρδος} \quad \text{Κωδικοποίησης}\right)$$

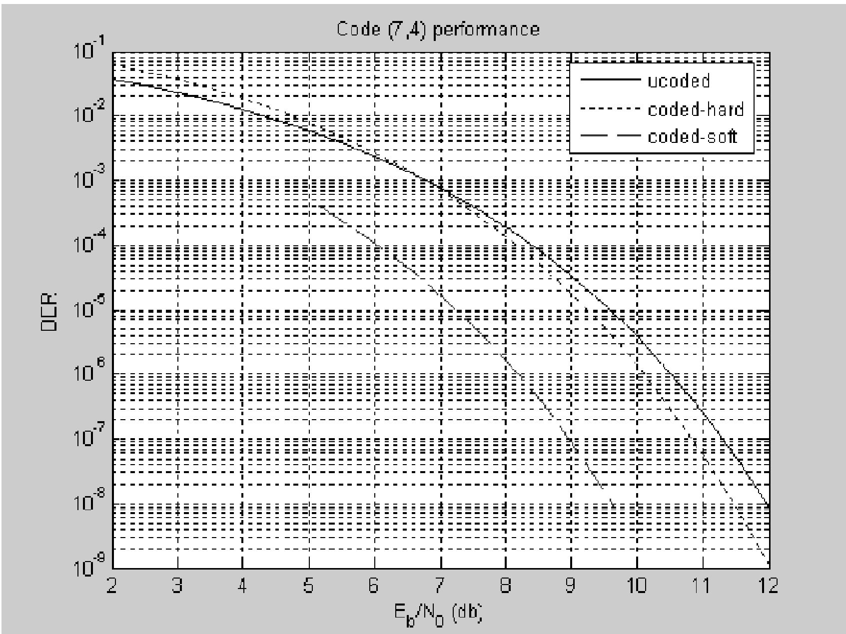
Με βάση τα συμπεράσματα που προέκυψαν από το παράδειγμα του κώδικα (5,2)



Από το διάγραμμα παρατηρείστε ότι η επίδοση της hard αποκωδικοποίησης στον κώδικα του παραδείγματος είναι αρνητική. Συγκεκριμένα απαιτεί μεγαλύτερη E_b για την ίδια πιθανότητα σφάλματος. Για το λόγο αυτό ο κώδικας (5,2) δεν χρησιμοποιείται στην πράξη για διαβίβαση μέσα από AWGN κανάλι.

Η Επίδοση της Soft Αποκωδικοποίησης είναι καλύτερη. Για τον υπολογισμό της καμπύλης έχουμε ελαττώσει το E_b/N_0 του uncoded κατά το κέρδος κωδικοποίησης που υπολογίστηκε ίσο με 2.37 dB. Η χάραξη της καμπύλης αυτής περιορίζεται μόνο στις υψηλές τιμές E_b/N_0 όπου και ισχύει ο υπολογισμός με βάση την ελάχιστη ευκλείδεια απόσταση d_{min}.

Στο επόμενο Σχήμα έχουν χαραχθεί οι επιδόσεις του κώδικα (7,4) τον οποίο θα συναντήσουμε σύντομα. Μπορείτε να διαπιστώσετε την αρτιότερη λειτουργία του κώδικα αυτού.



Τύποι Κωδίκων

Οι κώδικες καναλιού θεωρείται ότι διαχωρίζονται σε δύο μεγάλες κατηγορίες, τους *Κώδικες Block* και τους *Συνελικτικούς Κώδικες* (*Convolutional Codes*).

Στους κώδικες block μία λέξη από *k* bits δεδομένων, που καλούμε bits πληροφορίας, αντικαθίστανται από *n* κωδικά bits, από τα οποία ο δέκτης υπολογίζει τα *k* bits της πληροφορίας, ακόμα και αν έχει συμβεί πλήθος λαθών όχι μεγαλύτερο από ένα καθορισμένο πλήθος *t*.

Στους συνελικτικούς κώδικες τα bits του κώδικα αντιστοιχούν σε περισσότερες από μία λέξεις πληροφορίας και εκτείνονται σε μεγάλο μήκος.

Κώδικες Block

Από του κώδικες block θα εξετάσουμε μόνο τους *γραμμικούς δυαδικούς κώδικες*. Δηλαδή κώδικες στους οποίους τα κωδικά bits προκύπτουν με γραμμικές πράξεις από τα bits της πληροφορίας.

Οι πράξεις που εμφανίζονται στους δυαδικούς κώδικες είναι πολλαπλασιασμοί και προσθέσεις modulo 2.

Διαδικασία Κωδικοποίησης

Για τη δημιουργία των κωδικών λέξεων ενός κώδικα χρησιμοποιείται ο *Πίνακας Γεννήτορας* (*Generator Matrix*), **G**.

Για ένα *συστηματικό* (*systematic*) γραμμικό κώδικα block (*n*,*k*) ο πίνακας Γεννήτορας έχει διαστάσεις $k\times$n και η μορφή του είναι:

$$\mathbf{G}=\begin{bmatrix} 100\ldots0 & p_{11} & p_{12} & \cdots & p_{1,n-k} \\ 010\ldots0 & p_{21} & p_{22} & \cdots & p_{2,n-k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 000\ldots1 & p_{k1} & p_{k2} & \cdots & p_{k,n-k} \end{bmatrix}$$

Για τον υπολογισμό της κωδικής λέξης οποιασδήποτε από τις 2^k λέξεις με τα k bits πληροφορίας, οργανώνεται το διάνυσμα πληροφορίας **d**

$$\mathbf{d}=\left[d_1,\quad d_2,\quad d_3,\quad \cdots,\quad d_k\right]$$

και τα bits της αντίστοιχης κωδικής λέξης δίνονται οργανωμένα στο διάνυσμα **c** το οποίο προκύπτει:

$$\mathbf{c}=\mathbf{d}\cdot\mathbf{G}$$

Ή αναλυτικά

$$\left[c_1, c_2, ..., c_n\right]=\left[d_1, d_2, ..., d_k\right]\begin{bmatrix} 100\ldots0 & p_{11} & p_{12} & \cdots & p_{1,n-k} \\ 010\ldots0 & p_{21} & p_{22} & \cdots & p_{2,n-k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 000\ldots1 & p_{k1} & p_{k2} & \cdots & p_{k,n-k} \end{bmatrix}$$

Μπορείτε να παρατηρήσετε ότι ο Πίνακας Γεννήτορας γράφεται:

$$\mathbf{G}=\left[\mathbf{I}_k \mid \mathbf{P}\right]_{k\times n}$$

δηλαδή ο **G** αποτελείται από ένα μοναδιαίο Πίνακα $k\times k$ σε συνδυασμό με τον Πίνακα **P** διαστάσεων $k\times n\text{-}k$. Η δομή αυτή του G εξασφαλίζει καταρχήν τη συστηματικότητα του κώδικα, δηλαδή ότι τα k bits της πληροφορίας θα βρίσκονται στην αρχή της αντίστοιχης κωδικής λέξης.

Τα υπόλοιπα $n\text{-}k$ bits της κωδικής λέξης υπολογίζονται με βάση τον Πίνακα **P**. Τα bits αυτά είναι εκείνα που θα εξασφαλίσουν όλη την ικανότητα διόρθωσης του κώδικα και η επιλογή των στοιχείων του **P** είναι εξαιρετικά κρίσιμη για την ικανότητα αυτή.

Εκτός από ελάχιστες εξαιρέσεις δεν υπάρχει διαδικασία κατάλληλης επιλογής των στοιχείων του Πίνακα **P**.

Παράδειγμα

Για τον κώδικα (7,4) ο πίνακας **P** και ο Πίνακας Γεννήτορας **G** δίνεται πιο κάτω.

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Και ο Πίνακας με τις 2^4=16 λέξεις πληροφορίας και τις αντίστοιχες κωδικές λέξεις που προκύπτουν με βάση το γεννήτορα δίνονται στη συνέχεια.

0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	1
0	0	1	1	0	0	1	1	1	0	0
0	1	0	0	0	1	0	0	1	0	1
0	1	0	1	0	1	0	1	0	1	0
0	1	1	0	0	1	1	0	1	1	0
0	1	1	1	0	1	1	1	0	0	1
1	0	0	0	1	0	0	0	1	1	0
1	0	0	1	1	0	0	1	0	0	1
1	0	1	0	1	0	1	0	1	0	1
1	0	1	1	1	0	1	1	0	1	0
1	1	0	0	1	1	0	0	0	1	1
1	1	0	1	1	1	0	1	1	0	0
1	1	1	0	1	1	1	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1

Το Κωδικό Βιβλίο ενός Κώδικα (7,4)

<u>Ικανότητες των Γραμμικών Κωδίκων Block για Ανίχνευση και Διόρθωση.</u>

Βάρος, W μιας κωδικής λέξης καλείται το πλήθος των άσσων που περιέχει αυτή. Ελάχιστο βάρος, $W_{\min}$ του κώδικα είναι η ελάχιστη τιμή του βάρους για τον κώδικα αν παραλείψουμε την κωδική λέξη με όλο μηδενικά.

Αποδεικνύεται ότι:

$$d_{\min}^{H} = W_{\min}$$

Το πλήθος των σφαλμάτων, t, που μπορεί να διορθώσει ένα κώδικας block ισούται με τον μεγαλύτερο ακέραιο που δεν υπερβαίνει το μισό της ελάχιστης απόστασης Hamming του κώδικα ελαττωμένης κατά ένα.

$$t = \left\lfloor \frac{d_{\min}^{H} - 1}{2} \right\rfloor$$

Το πλήθος των σφαλμάτων, t', που μπορεί να ανιχνεύσει ένα κώδικας block ισούται με την ελάχιστη απόστασης Hamming του κώδικα ελαττωμένη κατά ένα.

$$t' = d_{\min}^{H} - 1$$

Αποκωδικοποίηση Κώδικα Block

<u>1. Ανίχνευση Λάθους στη Λέξη Λήψης</u>

Πίνακας Ελέγχου της Ισοτιμίας (*Parity Check Matrix*) καλείται ο πίνακας **H**:

$$\mathbf{H} = \left[\mathbf{P}^{T} \mid \mathbf{I}_{n-k} \right]_{(n-k)\times n} = \begin{bmatrix} p_{11} & p_{21} & \cdots & p_{k1} & 100...0 \\ p_{12} & p_{22} & \cdots & p_{k2} & 010...0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{1,n-k} & p_{2,n-k} & \cdots & p_{k,n-k} & 000...1 \end{bmatrix}$$

Ο πίνακας ελέγχου της ισοτιμίας μπορεί να χρησιμοποιηθεί για την επαλήθευση αν η λέξη λήψης, **r** ανήκει στον κώδικα καναλιού που χρησιμοποιείται.

Πιο συγκεκριμένα ορίζουμε ως *Σύνδρομο Λάθους* (*Error Syndrome*), **s**, το διάνυσμα με *n-k* στοιχεία

$$\mathbf{s} = \mathbf{r}\mathbf{H}^T = \mathbf{r}\left[\frac{\mathbf{P}}{\mathbf{I}_{n-k}}\right]$$

Αποδεικνύεται ότι όταν η λέξη λήψης **r** είναι κωδική λέξη το σύνδρομο **s** έχει όλες τις *n-k* συνιστώσες του ίσες με μηδέν.

Αντίθετα αν μία ή περισσότερες από τις *n-k* συνιστώσες του διανύσματος αποτελέσματος ισούται με 1 τότε η **r** δεν είναι κωδική λέξη και επομένως έχουν συμβεί λάθη κατά τη διαβίβαση.

Θεωρείστε ότι η λέξη λήψης **r** είναι το άθροισμα της κωδικής λέξης **c** που έστειλε ο πομπός και μιας λέξης σφαλμάτων **e** που καλείται *Ίχνος Σφάλματος*.

r = **c**+**e**

Τότε το σύνδρομο λάθους **s**

$$\mathbf{s} = \mathbf{r}\mathbf{H}^T = \left(\mathbf{c}+\mathbf{e}\right)\mathbf{H}^T = \mathbf{c}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T \Rightarrow$$
$$\mathbf{s} = \mathbf{e}\mathbf{H}^T$$

Όταν δηλαδή το σύνδρομο είναι μη μηδενικό, οι συνιστώσες του σχετίζονται με τα λάθη που έγιναν κατά τη διαβίβαση και επομένως μπορεί να χρησιμοποιηθεί για τη διόρθωση του λάθους. Η τεχνική αυτή είναι πιο ταχεία από εκείνη της αναζήτησης της κωδικής λέξης που απέχει ελάχιστα από την **r.**

Παράδειγμα

Θεωρείστε τον Πίνακα Γεννήτορα του κώδικα (7,4) που συναντήσαμε σε προηγούμενο Παράδειγμα.

$$\mathbf{G} = \left[\begin{array}{cccc|ccc} \multicolumn{4}{c}{\overbrace{\hphantom{1\;0\;0\;0}}^{\mathbf{I}_4}} & \multicolumn{3}{c}{\overbrace{\hphantom{1\;1\;0}}^{\mathbf{P}}} \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right]$$

Ο Πίνακας ελέγχου της ισοτιμίας του κώδικα θα είναι:

$$\mathbf{H} = \left[\mathbf{P}^T \mid \mathbf{I}_{n-k}\right] = \left[\begin{array}{cccc|ccc} \multicolumn{4}{c}{\overbrace{\hphantom{1\;1\;0\;1}}^{\mathbf{P}^T}} & \multicolumn{3}{c}{\overbrace{\hphantom{1\;0\;0}}^{\mathbf{I}_3}} \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Αν η λέξη λήψης **r** είναι **r**=[0 1 0 0 0 1 0]

Το σύνδρομο λάθους **s** υπολογίζεται:

$$\mathbf{s} = \mathbf{r}\mathbf{H}^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 111 \end{bmatrix}$$

Επομένως η λέξη **r** έχει υποστεί ένα τουλάχιστον λάθος.

2. Διόρθωση Λάθους στη Λέξη Λήψης

Οι τεχνικές διόρθωσης που θα αναφέρουμε εδώ είναι τεχνικές hard αποκωδικοποίησης.

2α. Διόρθωση Λάθους στη Λέξη Λήψης όταν ο Κώδικας έχει Δυνατότητα Διόρθωσης ενός Σφάλματος.

Όπως έχουμε ήδη αναφέρει το σύνδρομο λάθους **s** ισούται με

$$\mathbf{s} = \mathbf{e}\mathbf{H}^T$$

όπου **e** είναι το ίχνος σφάλματος (error pattern). Ειδικά για τους κώδικες διόρθωσης ενός σφάλματος η **e** έχει μία μόνο συνιστώσα ίση με 1. Αν η συνιστώσα αυτή είναι η *i*-οστή, τότε

$$\mathbf{s} = \mathbf{e}\mathbf{H}^T = i \text{ γραμμή του } \mathbf{H}^T$$

Όταν οι γραμμές του $\mathbf{H}^T$ είναι όλες διαφορετικές μεταξύ τους, από το σύνδρομο μπορούμε να βρούμε τον αριθμό της γραμμής που είναι ίδια με το σύνδρομο και στη συνέχει να κάνουμε τη διόρθωση του λάθους.

Παράδειγμα

Θυμηθείτε το τελευταίο παράδειγμα με λέξη λήψης

r=[0 1 0 0 0 1 0]

και σύνδρομο που υπολογίζεται ως:

$$\mathbf{s} = \mathbf{r}\mathbf{H}^T = \begin{bmatrix}0\ 1\ 0\ 0\ 0\ 1\ 0\end{bmatrix}\begin{bmatrix}1 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 1\\ 1 & 1 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} = \begin{bmatrix}111\end{bmatrix}$$

Παρατηρείστε ότι το σύνδρομο είναι ίδιο με την τέταρτη γραμμή του $\mathbf{H}^T$ επομένως το λάθος έχει συμβεί στην τέταρτη στήλη της λέξης λήψης, άρα η αντίστοιχη κωδική λέξη **c** είναι:

c=[0 1 0 1 0 1 0]➔ **d**=[0 1 0 1]

Οι γραμμικοί κώδικες block με δυνατότητα διόρθωσης ενός λάθους, καλούνται *Κώδικες Hamming* από το όνομα του ερευνητή που τους πρότεινε.

Στους κώδικες Hamming, αντίθετα από ότι συμβαίνει με τους υπόλοιπους κώδικες, υπάρχει καθορισμένος αλγόριθμος για τη σχεδίαση του Πίνακα **P** και επομένως του κώδικα γεννήτορα **G**.

Σχεδιασμός Κώδικα Hamming για Δοσμένο *k*

Καταρχήν θυμηθείτε την παρατήρηση που κάναμε στη διόρθωση λέξεων λήψης όταν ο κώδικας έχει δυνατότητα διόρθωσης ενός σφάλματος.

«Αν το λάθος βρίσκεται στην *i*-οστή στήλη του ίχνους σφάλματος, τότε το σύνδρομο ισούται με την *i*-οστή γραμμή του Πίνακα $\mathbf{H}^T$»

Για να αξιοποιήσουμε στη διόρθωση αυτή την ιδιότητα, ο Πίνακας $\mathbf{H}^T$ πρέπει να έχει διαφορετικές μεταξύ τους γραμμές.

$$\mathbf{H} = \begin{bmatrix}\mathbf{P}^T \mid \mathbf{I}_{n-k}\end{bmatrix}_{(n-k)\times n} = \begin{bmatrix} p_{11} & p_{21} & \cdots & p_{k1} & 100...0\\ p_{12} & p_{22} & \cdots & p_{k2} & 010...0\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ p_{1,n-k} & p_{2,n-k} & \cdots & p_{k,n-k} & 000...1\end{bmatrix}$$

$$\mathbf{H} = \left[\mathbf{P}^T \mid \mathbf{I}_{n-k}\right]_{(n-k)\times n} = \begin{bmatrix} p_{11} & p_{21} & \cdots & p_{k1} & 100...0 \\ p_{12} & p_{22} & \cdots & p_{k2} & 010...0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{1,n-k} & p_{2,n-k} & \cdots & p_{k,n-k} & 000...1 \end{bmatrix}$$

Παρατηρείστε τη μορφή του **H** που δίδεται πιο πάνω. Για να γίνει δυνατός ο στόχος για *n* διαφορετικές στήλες, με *n-k* στοιχεία η κάθε μια, χωρίς καμία από αυτές να έχει όλες τις συνιστώσες μηδέν, πρέπει να ισχύει

$$2^{n-k} - 1 \geq n \Leftrightarrow n \geq k + \log_2(n+1)$$

Από την τιμή του *k* προσδιορίζεται καταρχήν ο μικρότερος ακέραιος *n* που επαληθεύει τη σχέση, και στη συνέχεια κατασκευάζουμε τον αντίστοιχο Πίνακα **H** ως κάτωθι.

1. Παραλείπουμε τον συνδυασμό με *n-k* μηδενικά

2. Από τους 2^{n-k} -1 υπόλοιπους συνδυασμούς χρησιμοποιούμε τους *n-k* που έχουν μόνο ένα 1, για να κατασκευάσουμε τις *n-k* τελευταίες στήλες του **H** (το τμήμα του μοναδιαίου Πίνακα).

2. Χρησιμοποιούμε τους *k* από τους υπόλοιπους συνδυασμούς, φροντίζοντας να μη συμπεριλάβουμε το συνδυασμό με τα *n-k* μηδενικά, για να συμπληρώσουμε τις πρώτες *k* στήλες του **H**, δηλαδή τον Πίνακα $\mathbf{P}^T$.

3. Με τη βοήθεια του **P** κατασκευάζεται ο Πίνακας Γεννήτορας **G**.

Παράδειγμα

Να σχεδιάσετε έναν Κώδικα Hamming με μήκος λέξης πληροφορίας k=5. Φροντίστε να χρησιμοποιήσετε τη μικρότερη τιμή του n.

Απάντηση

Υπολογίζουμε τη μικρότερη τιμή του n που εξασφαλίζει τη σχέση

$$n \geq k + \log_2(n+1) \Leftrightarrow n \geq 5 + \log_2(n+1)$$

Η μικρότερη τιμή του n που επαληθεύει την πιο πάνω σχέση είναι n=9 και ο κώδικας θα είναι (9,5)

Ο Πίνακας **H** ορίζεται ως:

$$\mathbf{H} = \left[\mathbf{P}^T \mid \mathbf{I}_4\right]_{4\times 9} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

και ο Πίνακας γεννήτορας, **G**, είναι:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Ο πομπός, λοιπόν, για να στείλει μια λέξη πληροφορίας, έστω την

d=[1 0 0 1 0]

θα υπολογίσει και θα αποστείλει την αντίστοιχη κωδική λέξη **c**

c=d·G=[1 0 0 1 0 1 0 1 0]

Υποθέστε ότι στο δέκτη φθάνει η λέξη, δηλαδή έχει συμβεί 1 σφάλμα στη δεύτερη στήλη.

r=[1 1 0 1 0 1 0 1 0]

Υπολογίζεται το σύνδρομο λάθους **s**

$$\mathbf{s} = \mathbf{r}\mathbf{H}^T = [0101]$$

Όπως διαπιστώνουμε [0 1 0 1] είναι η δεύτερη στήλη του **H**.

2β. Διόρθωση Λάθους στη Λέξη Λήψης όταν ο Κώδικας έχει Δυνατότητα Διόρθωσης Περισσοτέρων από ένα Σφάλματα. Χρήση “Look up Table Decoder”

Για ένα κώδικα block (n,k) με ικανότητα διόρθωσης t>1 σφαλμάτων η διόρθωση των σφαλμάτων γίνεται πολύ πιο δύσκολα από ότι στους κώδικες Hamming.

Για ένα κώδικα block (n,k) με ικανότητα διόρθωσης t>1 σφαλμάτων καταστρώνεται πίνακας με όλες τις δυνατές τιμές του ίχνους σφάλματος που μπορούν να διορθωθούν και τα αντίστοιχα διανύσματα συνδρόμου σφάλματος.

Το πλήθος N αυτών των ιχνών είναι:

$$N = 1 + \binom{n}{1} + \ldots + \binom{n}{t}$$

Στον πίνακα αυτόν, *Πίνακα Επαλήθευσης* (*Look up Table*) περιλαμβάνετε επίσης το ίχνος με όλο μηδενικά, που αντιστοιχεί σε λήψη λέξης χωρίς σφάλμα.

Για να είναι δυνατή η διόρθωση θα πρέπει όλα αυτά τα ίχνη να δίνουν διαφορετικό σύνδρομο λάθους, δηλαδή πρέπει $2^{n-k} >= N$

Εν γένει ισχύει για κάθε κώδικα $2^{n-k}>N$, οπότε έχουμε τη δυνατότητα να διορθώσουμε και μερικά ακόμα ίχνη σφάλματος με t+1 άσσους, αρκεί να επιλέξουμε αυτά που δίδουν τα υπόλοιπα $2^{n-k}-N$ διαφορετικά σύνδρομα.

Η αποκωδικοποίηση γίνεται υπολογίζοντας το σύνδρομο **s** για τη λέξη λήψης, και από εκεί ευρίσκουμε το αντίστοιχο ίχνος σφάλματος το οποίο προσθέτουμε στην λέξη λήψης και λαμβάνεται η αντίστοιχη κωδική λέξη.

Στη συνέχεια δίνουμε ένα παράδειγμα αποκωδικοποίησης κώδικα block μέσω του Πίνακα των Επαλήθευσης . Δυστυχώς επειδή ο πιο απλός κώδικας διόρθωσης 2 σφαλμάτων δίνει πολύ μεγάλο πλήθος διαφορετικών συνδρόμων περιορίζουμε το παράδειγμα σε κώδικα διόρθωσης 1 σφάλματος.

Παράδειγμα

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Δίνεται ο Πίνακας **P** ενός κώδικα (6,3). Ο κώδικας έχει ικανότητα διόρθωσης ενός σφάλματος.

Ο Πίνακας έλεγχου της ισοτιμίας με τη μορφή $\mathbf{H}^T$ είναι:

$$\mathbf{H}^T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Επειδή το σύνδρομο έχει n-k=3 bits μπορούν να υπάρξουν 2^3=8 διαφορετικά σύνδρομα.

Σύνδρομα	Διορθώσιμα Ίχνη
0 0 0	0 0 0 0 0 0
0 0 1	0 0 0 0 0 1
0 1 0	0 0 0 0 1 0
0 1 1	0 1 0 0 0 0
1 0 0	0 0 0 1 0 0
1 0 1	0 0 1 0 0 0
1 1 0	1 0 0 0 0 0
1 1 1	1 0 0 0 0 1

Στον Πίνακα τοποθετήθηκαν τα 7 ίχνη με κανένα και έναν άσσο και ένα ίχνος με δύο άσσους., καθώς και τα αντίστοιχα σύνδρομα.

Ο πίνακας έχει διαταχθεί κατά αύξουσα σειρά του συνδρόμου ώστε να η προσπέλαση στο αντίστοιχο ίχνος να γίνεται με απλή διευθυνσιοδότηση.

Έστω ότι η λέξη λήψης είναι η

r=[0 1 0 1 1 0]

Υπολογίζεται το σύνδρομο του **r**

$$\boxed{\mathbf{s} = \mathbf{r}\mathbf{H}^T = 101}$$

Και από τον Πίνακα Επαλήθευσης βρίσκουμε ότι το σύνδρομο αυτό αντιστοιχεί στο ίχνος e=[0 0 1 0 0 0]. Οπότε η αντίστοιχη κωδική λέξη είναι η

c=r+e=[0 1 1 1 1 0]

Και η αντίστοιχη λέξη πληροφορίας είναι

d=[0 1 1]

Για κώδικα block με ικανότητα διόρθωσης πολλών σφαλμάτων ο πίνακας Επαλήθευσης γίνεται εξαιρετικά μεγάλος και η αναζήτηση του αντίστοιχου συνδρόμου εξαιρετικά πολύπλοκος.

Για παράδειγμα για τον κώδικα (200,175) ο Πίνακας Επαλήθευσης πρέπει να διαθέτει 2^{25} γραμμές με 225 bits η κάθε γραμμή (200 bit για το ίχνος και 25 για το σύνδρομο). Αυτό σημαίνει περίπου 7.6 10^9 bits!

Το ευτύχημα είναι ότι ένα υποσύνολο των κωδίκων block παρουσιάζει επιπλέον ιδιότητες που απλοποιούν σημαντικά την αποκωδικοποίηση. Αυτοί είναι οι *Κυκλικοί Κώδικες* που θα εξετάσουμε στη συνέχεια.

ΔΥΑΔΙΚΟΙ ΚΥΚΛΙΚΟΙ ΚΩΔΙΚΕΣ

Ένας block κώδικας, ο οποίος έχει επιπλέον την ιδιότητα, κάθε κυκλική περιστροφή (rotation) μιας κωδικής λέξης να είναι επίσης κωδική λέξη του κώδικα, καλείται *Κυκλικός Κώδικας* (*Cyclic Code*)

Οι κυκλικοί κώδικες παρουσιάζουν ιδιαίτερο ενδιαφέρον γιατί το σύνδρομο και οι κωδικές λέξεις υπολογίζονται με τη χρήση ψηφιακών ολισθητών με ανατροφοδότηση. Οι πράξεις αυτές παρουσιάζουν πολύ μικρότερη πολυπλοκότητα από τις πράξεις πινάκων. Επιπλέον αυτοί έχουν μια πολύ καλή μαθηματική δομή, η οποία κάνει δυνατή της σχεδίαση κωδίκων με χρήσιμες διορθωτικές ιδιότητες.

<u>Αλγεβρική Δομή των Κυκλικών Κωδίκων</u>

Σύμφωνα με τον ορισμό που δόθηκε πιο πάνω, αν η λέξη

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

είναι κωδική λέξη ενός (n,k) κώδικα C, τότε η λέξη:

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

είναι επίσης κωδική λέξη του κώδικα C.

Με βάση τον ορισμό αυτό και η λέξη

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

είναι επίσης κωδική λέξη του C.

Η ιδιότητα αυτή, όπως και οι τεχνικές κωδκοποίησης και αποκωδικοποίησης ενός κυκλικού κώδικα αποδίδεται τέλεια με τη χρήση των πολυωνύμων με δυαδικούς συντελεστές.

Έτσι τα bits της κωδικής λέξης

$$V=(\upsilon_0, \upsilon_1, \upsilon_2 \ldots, \upsilon_{n-2}, \upsilon_{n-1})$$

Μπορεί να θεωρηθούν ότι είναι οι συντελεστές του πολυωνύμου

$$V(x)=\upsilon_0+\upsilon_1 x + \upsilon_2 x^2+ \ldots + \upsilon_{n-2}x^{n-2} + \upsilon_{n-1}x^{n-1}).$$

Αποδεικνύεται ότι το πολυώνυμο της i-οστής αριστερόστροφης περιστροφής της κωδικής λέξη V, το $V^{(i)}(x)$

$$V^{(i)}(x)=\upsilon_{n-i}+\upsilon_{n-i+1}x+\ldots+\upsilon_0 x^i +\upsilon_1 x^{i+1} + \ldots + \upsilon_{n-i-1}x^{n-1}).$$

ισούται με το υπόλοιπο της διαίρεσης του $x^i\mathrm{V}(x)$ δια του x^n+1, δηλαδή ισχύει:

$$x^i\mathrm{V}(x)=q(x)(x^n+1) + V^{(i)}(x)$$

Πολυώνυμο Γεννήτορας ενός Κυκλικού Κώδικα.

Ένα πολυώνυμο $g(x)$ βαθμού n-k, το οποίο είναι διαιρέτης του x^n+1 μπορεί να δημιουργήσει τις κωδικές λέξεις ενός κυκλικού κώδικα (n,k). Το $g(x)$ καλείται Πολυώνυμο Γεννήτορας του κυκλικού κώδικα.

Αν $g(x)=g_0+ g_1x+\ldots+g_{n-k}x^{n-k}$, τότε $g_{n-k}=1$ και $g_0=1$. Δηλαδή $g(x)=1+ g_1x+\ldots+g_{n-k-1}x^{n-k-1}+x^{n-k}$

Κωδικοποίηση Δεδομένων με Κώδικα (n,k)

Οι κωδικές λέξεις προκύπτουν από τα 2^k πολυώνυμα πληροφορίας $D(x)$ και το πολυώνυμο γεννήτορα $g(x)$ με μια από τις ακόλουθες διαδικασίες.

Διαδικασία 1.

Το $V(x)$ προκύπτει ως

$$V(x) = D(x)g(x)$$

Για παράδειγμα το πολυώνυμο τρίτου βαθμού $g(x)=1+x+x^3$ είναι διαιρέτης του x^7+1. Με βάση το $g(x)$ κατασκευάζεται ο (7,4) κυκλικός κώδικας. Το κωδικό πολυώνυμο $V(x)$ για το διάνυσμα πληροφορίας (1 0 1 0) με πολυώνυμο $D(x)=1+x^2$ υπολογίζεται ως:

$$V(x) = D(x)g(x) =(1+x^2)(1+x+x^3)=1+x^2+x+x^3+x^3+x^5=1+x+x^2+x^5$$

Δηλαδή η αντίστοιχη κωδική λέξη είναι:

$$V =(1\ 1\ 1\ 0\ 0\ 1\ 0)$$

Ανάλογα βρίσκουμε ολόκληρο το codebook με τις 2^k κωδικές λέξεις

Διαδικασία 2.

Με τη διαδικασία αυτή ο κώδικας που προκύπτει είναι συστηματικός. Για δοσμένο το $D(x)$:

2α). Υπολογίζονται τα n-k bits ελέγχου της ισοτιμίας, $r_0, r_1, \ldots, r_{n-k-1}$ ως το πολυώνυμο-υπόλοιπο $r(x)$ της διαίρεσης $x^{n-k}D(x)$ δια του $g(x)$. Δηλαδή:

$$x^{n-k}D(x)=q(x)g(x)+r(x)$$

2β). Το κωδικό πολυώνυμο $V(x)$ προκύπτει ως

$$V(x) =r(x)+x^{n-k}D(x)$$

Για παράδειγμα, για το πολυώνυμο γεννήτορα τρίτου βαθμού $g(x)=1+x+x^3$ και το πολυώνυμο πληροφορίας $D(x)=1+x^2$ που είδαμε στο παράδειγμα της Διαδικασίας 1, η κωδική λέξη σε συστηματική μορφή προκύπτει ως εξής:

Καταρχήν υπολογίζεται το πολυώνυμο έλέγχου της ισοτιμίας ως το υπόλοιπο της διαίρεσης $x^3(1+x^2)=x^5+x^3$ διά του $1+x+x^3$.

$$\begin{array}{l|l} x^5+x^3 & x^3+x+1 \\ \hline x^5+x^3+x^2 & x^2 \\ \qquad\quad x^2 & \end{array}$$

Επομένως $r(x)=x^2 \Rightarrow r=(0\ 0\ 1)$.

$V(x)=x^2+x^3(1+x^2)=x^2+x^3+x^5 \Rightarrow V=(0\ 0\ 1\ 1\ 0\ 1\ 0)$

Εύκολα προκύπτει ότι το $g(x)$ διαιρεί ακριβώς τα κωδικά πολυώνυμα είτε αυτά ανήκουν σε συστηματικό ή σε μη συστηματικό κώδικα.

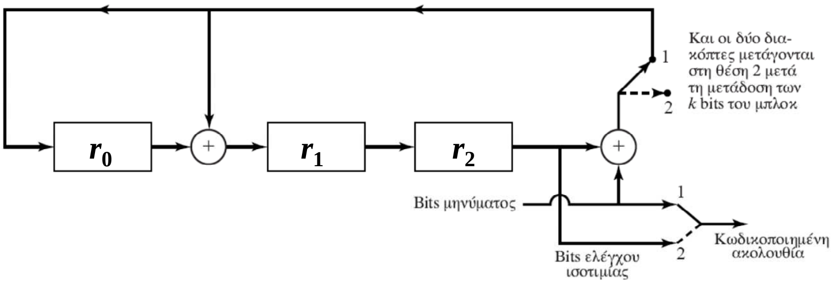
Στην επομένη σελίδα δίνουμε τις 16 κωδικές λέξεις για τους δύο κυκλικούς κώδικες που κατασκευάζονται με τις δύο διαδικασίες μέσω του πολυωνύμου γεννήτορα $g(x)=1+x+x^3$.

Κυκλικός Κώδικας (7,4) Πολυώνυμο Γεννήτορας $g(x)=1+x+x^3$.

D	$V(x)=D(x)g(x)$	$V(x)=r(x)+x^3D(x)$
0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 1	0 0 0 1 1 0 1	1 0 1 0 0 0 1
0 0 1 0	0 0 1 1 0 1 0	1 1 1 0 0 1 0
0 0 1 1	0 0 1 0 1 1 1	0 1 0 0 0 1 1
0 1 0 0	0 1 1 0 1 0 0	0 1 1 0 1 0 0
0 1 0 1	0 1 1 1 0 0 1	1 1 0 0 1 0 1
0 1 1 0	0 1 0 1 1 1 0	1 0 0 0 1 1 0
0 1 1 1	0 1 0 0 0 1 1	0 0 1 0 1 1 1
1 0 0 0	1 1 0 1 0 0 0	1 1 0 1 0 0 0
1 0 0 1	1 1 0 0 1 0 1	0 1 1 1 0 0 1
1 0 1 0	1 1 1 0 0 1 0	0 0 1 1 0 1 0
1 0 1 1	1 1 1 1 1 1 1	1 0 0 1 0 1 1
1 1 0 0	1 0 1 1 1 0 0	1 0 1 1 1 0 0
1 1 0 1	1 0 1 0 0 0 1	0 0 0 1 1 0 1
1 1 1 0	1 0 0 0 1 1 0	0 1 0 1 1 1 0
1 1 1 1	1 0 0 1 0 1 1	1 1 1 1 1 1 1

Υλοποίηση Κωδικοποιητή Κυκλικού Κώδικα

Το κύκλωμα ενός κωδικοποιητή για πολυώνυμο γεννήτορα $g(x)=1+x+x^3$ βαθμού 3 και για υλοποίηση συστηματικού κυκλικού κώδικα (7,4) δίνεται στο ακόλουθο σχήμα.



Το κύκλωμα αυτό, βασικά υπολογίζει το υπόλοιπο της διαίρεσης $x^3D(x)$ διά του $g(x)=1+x+x^3$ για μια λέξη πληροφορίας $D(x)$

Αυτό αποτελείται από ένα δεξιό κυκλικό ολισθητή (RSR) με τρία flip flop (F-F) και με ανατροφοδότηση τους 0-2 συντελεστές του $g(x)$, (1,1,0), δηλαδή στην είσοδο του r_0 και r_1 F-F.

Αρχικά οι δύο διακόπτες κάνουν επαφή με τη θέση 1, και τα bits μηνύματος εισέρχονται στο κανάλι και συγχρόνως ανατροφοδοτούνται προς τον RSR.

Όταν τα 4 bits της πληροφορίας έχουν εξαντληθεί, τα 3 bits ελέγχου της ισοτιμίας έχουν πλέον καθοριστεί και βρίσκονται ως υπόλοιπο στα F-F r_0, r_1 και r_2. Τότε απενεργοποιείται η πύλη AND και οι διακόπτες κλείνουν στη θέση 2. Έτσι τα τρία bits ελέγχου της ισοτιμίας οδεύουν προς το κανάλι, ενώ συγχρόνως καθαρίζεται ο RSR.

Πώς με τον τρόπο λειτουργίας που περιγράψαμε δημιουργείται το υπόλοιπο της διαίρεσης $x^3D(x)$:$g(x)$, δεν θα το δείξουμε, αλλά θα το παρακολουθήσουμε με ένα παράδειγμα.

Υποθέστε ότι αποστέλλεται η λέξη πληροφορίας $D=(d_0\ d_1\ d_2\ d_3)$ έχει τιμή (1 0 1 1).

Στον Πίνακα που ακολουθεί βλέπουμε πως διαμορφώνεται η τελική τιμή των r_0, r_1 και r_2

$D=(1\ 0\ 1\ 1)$, $g(x)=1+x+x^3$

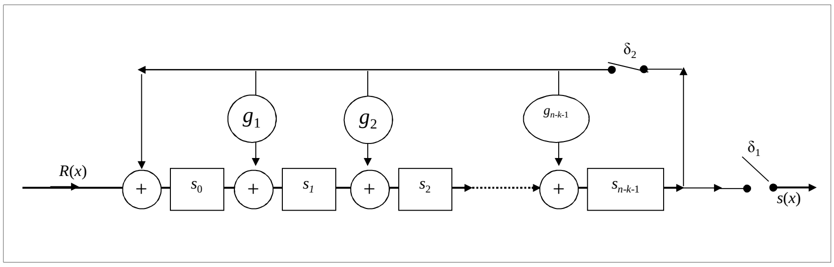
r_0	r_1	r_2	d_i	input= d_i+r_2
0	0	0	$d_3=1$	1
1	1	0	$d_2=1$	1
1	0	1	$d_1=0$	1
1	0	0	$d_0=1$	1
1	0	0		

Υπολογισμός του Συνδρόμου (*n*,*k*) Κυκλικού Κώδικα

Όπως έχουμε ήδη αναφέρει, ισχύει ότι το $g(x)$ διαιρεί ακριβώς τα κωδικά πολυώνυμα είτε αυτά ανήκουν σε συστηματικό είτε σε μη συστηματικό κώδικα.

Στο πιο κάτω σχήμα δίνεται το κύκλωμα ενός υπολογιστή συνδρόμου για τον κώδικα (n,k) με το πολυώνυμο γεννήτορα $g(x)=1+g_1x+\ldots+g_{n-k-1}x^{n-k-1}+x^{n-k}$. Είναι προφανής η ομοιότητα με το κύκλωμα του κωδικοποιητή.

Κύκλωμα Υπολογισμού Συνδρόμου (*n*,*k*) Κυκλικού Κώδικα.



Στο κύκλωμα αυτό είναι φανερή η ομοιότητα με το κύκλωμα κωδικοποιητή (n,k) κυκλικού κώδικα.

Το κύκλωμα αποτελείται από ένα καταχωρητή δεξιάς περιστροφής με ανατροφοδοτήσεις τους συντελεστές του $g(x)$.

Αρχικά ο καταχωρητής περιέχει μηδενικά, ο δ_1 είναι ανοικτός και ο δ_2 κλειστός.

Η λέξη λήψης $R(x)$ εισέρχεται στον καταχωρητή, ένα bit ανά clock. Όταν ολοκληρωθεί η είσοδος των bits της $R(x)$ έχει ολοκληρωθεί επίσης και ο υπολογισμός του συνδρόμου $s(x)$, το οποίο βρίσκεται στον καταχωρητή.

Τότε ο δ_2 ανοίγει (απενεργοποιούνται οι ανακυκλώσεις) και κλείνει ο δ_1 οπότε εξέρχεται το σύνδρομο και μηδενίζεται ο καταχωρητής.

Αποκωδικοποίηση (n,k) Κυκλικού Κώδικα

Όπως έχουμε ήδη αναφέρει, ισχύει ότι το $g(x)$ διαιρεί ακριβώς τα κωδικά πολυώνυμα είτε αυτά ανήκουν σε συστηματικό είτε σε μη συστηματικό κώδικα.

Αν $R(x)$ είναι το πολυώνυμο λήψης υπολογίζουμε το υπόλοιπο της διαίρεσης πολυωνύμων $R(x) : g(x)$, δηλαδή ένα πολυώνυμο $s(x)$ με βαθμό το πολύ n-k-1, το οποίο καλείται *πολυώνυμο συνδρόμου*.

Αν όλοι οι n-k συντελεστές του συνδρόμου είναι ίσοι με μηδέν, τότε η $R(x)$ είναι ένα από τα κωδικά πολυώνυμα του κώδικα, αλλιώς δεν είναι κωδικό πολυώνυμο.

Όταν το $R(x)$ είναι δεν είναι κωδικό πολυώνυμο, μπορεί να θεωρηθεί ως το άθροισμα ενός κωδικού πολυωνύμου, $V(x)$ και ενός πολυωνύμου ίχνους σφάλματος, $\varepsilon(x)$.

Δηλαδή:

$R(x)=V(x)+\varepsilon(x)$

Εύκολα προκύπτει ότι το σύνδρομο του $R(x)$ ισούται με το σύνδρομο του $\varepsilon(x)$. Όταν λοιπόν έχει συμβεί λάθος το σύνδρομο $s(x)$ δίνει δυνατότητα προσδιορισμού του $\varepsilon(x)$, και διόρθωσης του $R(x)$, ώστε να ληφθεί η $V(x)$

Μέχρι του σημείου αυτού οι ιδιότητες που περιγράψαμε για τους κυκλικούς κώδικες είναι εντελώς αντίστοιχες εκείνων των κωδίκων block.

Υπάρχει όμως μια επιπλέον ιδιότητα, η οποία δίνει τη δυνατότητα να ελαττωθεί σημαντικά η απαίτηση μνήμης και η πολυπλοκότητα του αποκωδικοποιητή.

Βασική Ιδιότητα Συνδρόμου Κυκλικού Κώδικα

Αν $s(x)$ είναι το σύνδρομο του ίχνους σφάλματος, τότε το σύνδρομο της κυκλικής περιστροφής του $\varepsilon(x)$, του $\varepsilon^{(1)}(x)$, έχει σύνδρομο ίσο με το υπόλοιπο της διαίρεσης του $xs(x)$ διά του $g(x)$.

Για παράδειγμα στον κώδικα (7,4) με πολυώνυμο γεννήτορα $g(x)=1+x+x^3$ το ίχνος σφάλματος $\varepsilon=[0\ 0\ 0\ 0\ 0\ 0\ 1]$ έχει σύνδρομο $s=[1\ 0\ 1]$. Το σύνδρομο του ίχ.σφ. $\varepsilon^{(1)}=[1\ 0\ 0\ 0\ 0\ 0\ 0]$ ισούται με το υπόλοιπο της διαίρεσης $xs(x)$ διά του $g(x)$, και είναι [1 0 0]

Προφανώς η ιδιότητα αυτή γενικεύεται ως:

Το σύνδρομο της κυκλικής περιστροφής του $\varepsilon(x)$, του $\varepsilon^{(i)}(x)$, έχει σύνδρομο ίσο με το υπόλοιπο της διαίρεσης του $x^i s(x)$ διά του $g(x)$.

Έτσι για το προηγούμενο παράδειγμα με $g(x)$= $1+x+x^3$, στο οποίο για ε=[0 0 0 0 0 0 1] s=[1 0 1], γενικεύεται και δίνει τον πιο κάτω Πίνακα:

ι	Ιχν. Σφαλ. ε	s: Σύνδρομο του ε	$x^i s(x)$	$x^i s(x)$ mod($g(x)$)
0	0 0 0 0 0 0 1	1 0 1	$1+x^2$	$1+x^2$
1	1 0 0 0 0 0 0	1 0 0	$x+x^3$	1
2	0 1 0 0 0 0 0	0 1 0	x^2+x^4	x
3	0 0 1 0 0 0 0	0 0 1	x^3+x^5	x^2
4	0 0 0 1 0 0 0	1 1 0	x^4+x^6	1+x
5	0 0 0 0 1 0 0	0 1 1	x^5+x^7	$x+x^2$
6	0 0 0 0 0 1 0	1 1 1	x^6+x^8	$1+x+x^2$

Αντί λοιπόν να αποθηκεύσουμε στο lookup table τα 7 ίχνη σφάλματος με τα αντίστοιχα σύνδρομα, αρκεί να αποθηκευθεί μόνο το πρώτο σύνδρομο. Τα υπόλοιπα 6 ίχνη και τα σύνδρομά τους υπολογίζονται με απλούς υπολογισμούς κατά το χρόνο της αποκωδικοποίηση της $R(x)$.

Για κώδικες που διορθώνουν 2 ή περισσότερα σφάλματα η οικονομία είναι ακόμα μεγαλύτερη.

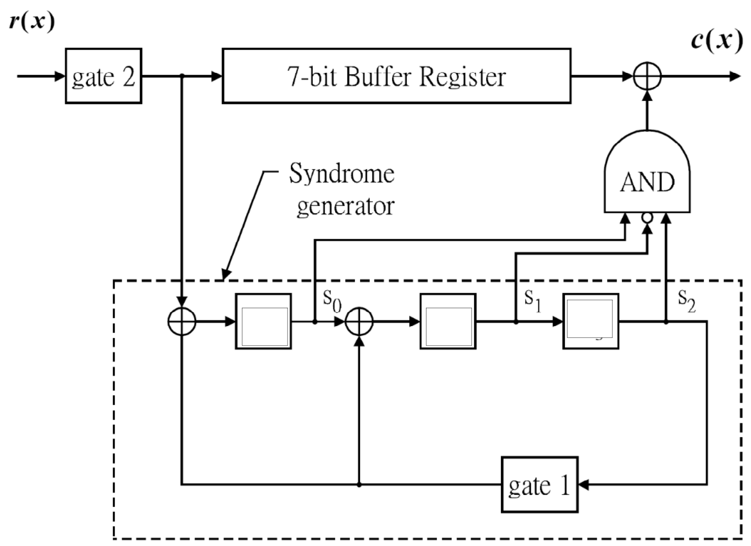
Αλλά και ο αλγόριθμος αποκωδικοποίησης, λόγω της βασικής ιδιότητας του συνδρόμου των κυκλικών κωδίκων, απλοποιείται σημαντικά.

Για παράδειγμα, για τον (7,4) κώδικα με τον Πίνακα της προηγούμενης σελίδας, η αποκωδικοποίηση γίνεται με τον πιο κάτω αλγόριθμο.

Αλγόριθμος Αποκωδικοποίησης του (7,4) κώδικα

- Θέσε $s_1(x)$=101 (Θυμηθείτε ότι το σύνδρομο αυτό αντιστοιχεί στο ίχν.σφαλ. 0000001)
- Υπολόγισε το σύνδρομο $s(x)$ για τη λέξη λήψης $R(x)$.
- for i=1:7
- If $s(x)=s_1(x)$
- i bit της $R(x)$ λάθος , Διόρθωσε R(x)
- else
- Υπολόγισε σύνδρομο $s(x)$ του $xs(x)$
- endif
- continue

Κύκλωμα Υλοποίησης του Αλγ. Αποκωδ. του (7,4) Κώδικα.



ΚΩΔΙΚΕΣ BCH

Οι BCH κώδικες αποτελούν ένα υποσύνολο των κυκλικών κωδίκων, και παρουσιάζουν βασικά πλεονεκτήματα σε σχέση με τελευταίους. Το μεγαλύτερο πλεονέκτημά τους είναι οι ειδικές ιδιότητες που διαθέτουν στη δομή τους, ακόμα και όταν αυτοί διορθώνουν πολλά λάθη. Οι ιδιότητες αυτές έχουν ως αποτέλεσμα ο αντίστοιχος αποκωδικοποιητής να υλοποιείται με πολύ απλά κυκλώματα.

Για το λόγο αυτό, αρχικά οι κώδικες αυτοί χρησιμοποιήθηκαν κατά κόρο σε διαστημικά οχήματα, και πιο πρόσφατα στην τηλεφωνία κινητών.

Ένα ακόμη πλεονέκτημά τους είναι το ότι για παραμέτρους μήκους n και ικανότητα διόρθωσης σφαλμάτων t (που ακολουθούν δοσμένους περιορισμούς επιλογής), υπάρχει μέθοδος που υπολογίζει το πολυώνυμο γεννήτορα $g(x)$.

Συγκεκριμένα οι κωδικές λέξεις στους κώδικες BCH έχουν πάντα μήκος n που δίνεται από τη σχέση

$$n = 2^m - 1$$

όπου m ακέραιος με $m>2$.

Δηλαδή στους BCH ισχύει n=7, 15, 31, 63 κλπ.

Αν τώρα για δοσμένη τιμή του n της πιο πάνω μορφής επιθυμούμε να κατασκευάσουμε κώδικα με ικανότητα διόρθωσης t σφαλμάτων, τότε υπάρχει αλγόριθμος για να προσδιοριστεί το $g(x)$ του κώδικα (n,k) με ικανότητα διόρθωση t σφαλμάτων, επιτυγχάνοντας μήκος k

$$k >= n-mt$$

Παράδειγμα

Σε ένα κώδικα BCH με n=31 επιθυμούμε t=5. Ποια θα είναι η μικρότερη τιμή του του ρυθμού του κώδικα;

n=31 ➔ m=5. Οπότε $k>=n-mt=31-5\times5=6$

και επομένως ο ρυθμός $\rho=k/n>=6/31$

Όταν όμως προσδιοριστεί το $g(x)$ προκύπτει κώδικας (31,11) με ρυθμό 11/31, σχεδόν διπλάσιος από το κάτω όριο που υπολογίζει η θεωρία

Ο υπολογισμός του πολυωνύμου $g(x)$ ξεπερνά τους στόχους της παρουσίασης αυτής.

Στον Πίνακα που ακολουθεί δίνονται οι διαστάσεις (n,k) και οι ικανότητα t για κώδικες BCH με n=7, 15, 31 και 63. Επίσης δίνονται και τα αντίστοιχα πολυώνυμα γεννήτορες $g(x)$ με τους συντελεστές κωδικοποιημένους σε 8-δικά ψηφία. Για παράδειγμα ο κώδικας (15,5) διορθώνει t=3 λάθη, και οι συντελεστές του $g(x)$ είναι:

2467➔10,100,110,111➔$g(x)=x^{10}+x^8+x^5+x^4+x^2+x+1$

ΠΙΝΑΚΑΣ 9.1 ΣΥΝΤΕΛΕΣΤΕΣ ΠΟΛΥΩΝΥΜΟΥ-ΓΕΝΝΗΤΟΡΑ ΚΩΔΙΚΩΝ BCH

n	k	t	$g(p)$
7	4	1	12
15	11	1	23
	7	2	721
	5	3	2467
31	26	1	45
	21	2	3551
	16	3	107657
	11	5	5423325
	6	7	313365047
63	57	1	103
	51	2	12471
	45	3	1701317
	39	4	166623567
	36	5	1033500423
	30	6	157464165547
	24	7	17323260404441
	18	10	1363026512351725
	16	11	6331141367235453
	10	13	472622305527250155
	7	15	5231045543503271737

ΕΠΙΒΡΑΧΥΜΕΝΟΙ ΚΥΚΛΙΚΟΙ BCH ΚΩΔΙΚΕΣ (SHORTENED CODES)

Όπως έχουμε ήδη αντιληφθεί, για μια τρέχουσα εφαρμογή, στην οποία χρειαζόμαστε ένα κώδικα με δεδομένες τιμές *n*,*k*, *t* δεν βρίσκουμε πάντα κωδικό πολυώνυμο, ειδικά αν δεν πρόκειται για BCH κώδικα.

Για παράδειγμα αν επιθυμούμε να αποστείλουμε δεδομένα πληροφορίας με k=8 bits και προσπαθήσουμε να βρούμε κώδικα BCH με δυνατότητα διόρθωσης t=2 σφάλματα δεν θα το επιτύχουμε.

Καθώς εκτός από του BCH δύσκολα προσδιορίζονται τα πολυώνυμα γεννήτορες, το πρόβλημα λύνεται ακολουθώντας το εξής τέχνασμα:

Επιλέγουμε έναν BCH κώδικα με t=2 και με $k \geq 8$. Από τον πίνακα του BCH εντοπίζουμε τον (31,21). Τον κώδικα αυτό τον υλοποιούμε με συστηματική μορφή.

Με το κώδικα αυτό όμως στέλνουμε λέξεις πληροφορίας με 8 bits που συμπληρώνεται μέχρι τα 21 με 13 μηδενικά. Έτσι προκύπτει κωδική λέξη με 31 bits της οποίας τα 13 τελευταία bits είναι όλα μηδενικά.

Τα 13 αυτά bits αποκόπτονται και αποστέλλονται τα υπόλοιπα 31-13=18 bits

Ο δέκτης στη λέξη λήψης των 18 bits προσθέτει 13 μηδενικά, και αποκωδικοποιεί τη λέξη λήψης των 31 bits που προκύπτει.

Με την τεχνική που περιγράψαμε, στην ουσία ο κώδικας που χρησιμοποιούμε είναι (18,8) με ικανότητα διόρθωσης 2 σφαλμάτων.

Οι κώδικες αυτοί μπορούν να κατασκευαστούν εύκολα και καλούνται *Επιβραχυμένοι Κώδικες* (*Shortened Codes*)**.**

Γενικά οι κώδικες αυτοί κατασκευάζονται από BCH κώδικα (n,k) σε (n-i,k-i) κώδικα.

ΚΩΔΙΚΕΣ ΣΥΜΠΛΕΞΗΣ

(INTERLEAVING CODES)

Για τη διόρθωση τυχαίων σφαλμάτων και καταιγισμών καθορισμένου μήκους q χρησιμοποιούμε την τεχνική της σύμπλεξης.

Σφάλμα καταιγισμού μήκους q καλούμε ίχνος σφάλματος στο οποίο τα μη μηδενικά bits συγκεντρώνονται σε q διαδοχικές θέσεις. Για παράδειγμα στο ίχνος σφάλματος 0 0 0 1 0 0 1 1 0 1 0 0 0, υπάρχει καταιγισμός μήκους q=7.

Τα σφάλματα καταιγισμού εμφανίζονται όταν ο θόρυβος είναι κρουστικός και όχι Gaussian. Για να αντιμετωπίσουμε τους καταιγισμούς σφαλμάτων δεν είναι απαραίτητο να χρησιμοποιηθεί κώδικας με t=q. Ένας τέτοιος κώδικας θα έχει εξαιρετικά μικρό ρυθμό.

Αντίθετα μπορούμε να χρησιμοποιήσουμε έναν κώδικα με $t<q$ αρκεί στον πίνακα αναζήτησης να καταχωρούνται μόνο τα ίχνη με καταιγισμούς μήκους *q*.

Η χρήση της τεχνικής της σύμπλεξης (interleaving) εξασφαλίζει επίσης μια αποτελεσματική αντιμετώπιση καταιγισμών σε συνδυασμό με την αντιμετώπιση των τυχαίων σφαλμάτων.
Στην τεχνική σύμπλεξης χρησιμοποιείται κώδικας με ικανότητα διόρθωσης ενός ή δύο σφαλμάτων για την κατασκευή ενός πιο σύνθετου κώδικα που διορθώνει καταιγισμό μήκους q καθώς και τυχαία σφάλματα που πιθανόν να συμβούν παράλληλα με τους καταιγισμούς.

Για παράδειγμα αν επιθυμούμε να αντιμετωπίσουμε καταιγισμό μήκους q=5, θεωρούμε τον κώδικα (15,7) με t=2 και πολυώνυμο γεννήτορα κωδικοποιημένο σε 721. Με τον κώδικα αυτό κωδικοποιούνται q=5 λέξεις πληροφορίας μήκους συνολικά 35 bits σε 5 κωδικές λέξεις των 15 bits η κάθε μία και αποθηκεύονται σε διαδοχικές λέξεις στη μνήμη όπως στο ακόλουθο σχήμα.

$1^η$ κωδ. λέξη	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$2^η$ κωδ. λέξη	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
$3^η$ κωδ. λέξη	31	32	33	34	35	36	37	38	…					…	45
$4^η$ κωδ. λέξη	46	47	…											…	60
$5^η$ κωδ. λέξη	61	62	…									…	73	74	75

Μετά την αποθήκευση των 75 συνολικά κωδικών bits στον πίνακα γίνεται η εκπομπή τους, ακολουθώντας όμως τώρα τη σειρά που δείχνουν τα βέλη, δηλαδή 1,16,31, …. Στην ακολουθία αυτή λαμβάνει ο δέκτης και τοποθετεί σε παρόμοιο πίνακα μνήμης.

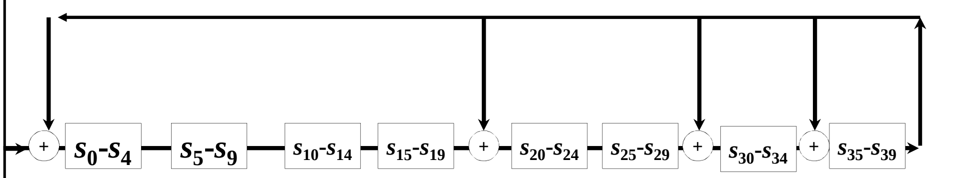
Αν κατά την πιο πάνω διαβίβαση συμβεί καταιγισμός σφαλμάτων, ή ακόμα και τυχαία σφάλματα, όπως υποδεικνύεται στον πίνακα που ακολουθεί, τα σφάλματα αυτά διορθώνονται αφού κάθε γραμμή του πίνακα είναι λέξη που προέρχεται από τον κώδικα (15,7) με ικανότητα διόρθωσης t=2

1η κωδ. λέξη	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2η κωδ. λέξη	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
3η κωδ. λέξη	31	32	33	34	35	36	37	38	…					…	45
4η κωδ. λέξη	46	47	…											…	60
5η κωδ. λέξη	61	62	…									…	73	74	75

Αν είναι $g(x)$ το πολυώνυμο γεννήτορα του αρχικού (n,k) κώδικα που χρησιμοποιήθηκε για τη δημιουργία του κώδικα σύμπλεξης, τότε ο τελευταίος είναι επίσης κυκλικός κώδικας με πολυώνυμο γεννήτορα $g_{in}(x)=g(x^{\lambda})$, όπου λ είναι το πλήθος των γραμμών του Πίνακα. Έτσι για το πιο πάνω παράδειγμα με αρχικό κώδικα (15,7) το πολυώνυμο γεννήτορας κωδικοποιείται ως 721, δηλαδή είναι το $x^8+ x^7+ x^6+ x^4+1$ ή καλύτερα $g(x)=1+x^4+ x^6+ x^7+ x^8$.

Επομένως ο κώδικας σύμπλεξης είναι ένας κυκλικός κώδικας (75,35) με πολυώνυμο γεννήτορα $g_{in}(x)= + 1 + x^{20} + x^{30} + x^{35} +x^{40}$.

Επομένως μπορεί να υλοποιηθεί το κύκλωμα του συνδρόμου, όπως στο ακόλουθο σχήμα.



Παράδειγμα

Σε ένα κανάλι επικοινωνίας εμφανίζεται κρουστικός θόρυβος διάρκειας το πολύ 10 μsec και με ελάχιστη χρονική απομάκρυνση μεταξύ των στιγμών θορύβου 140 μsec. Στο κανάλι αυτό δεν υπάρχει Gaussian θόρυβος και διαβιβάζονται μέσα από αυτό δεδομένα με ρυθμό 1bit/μsec. Σχεδιάστε ένα σχήμα κώδικα σύμπλεξης με ρυθμό μεγαλύτερο από 4/7 που να αντιμετωπίζει το πρόβλημα του καταιγισμού σφαλμάτων.