

Τμήμα
Μηχανικών
Πληροφορικής τ.ε.

Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

Πτυχιακή Εργασία

**Θέμα: Υλοποίηση των βασικών συνιστωσών του
πρωτοκόλλου DVB.**

Σπουδαστής:
Γκούβας Κωνσταντίνος

Επιβλέπων Καθηγητής:
Τσακανίκας Βασίλειος

Αντίρριο 2014

ΠΕΡΙΛΗΨΗ

Σκοπός της πτυχιακής εργασίας είναι η μελέτη του πρωτοκόλλου DVB και η ανάπτυξη γρήγορων αλγορίθμων για την υλοποίηση των βασικότερων συνιστωσών του. Δημιουργείται πλατφόρμα προσομοίωσης για την παρατήρηση των βασικότερων παραμέτρων της ασύρματης μετάδοσης.

Ειδικότερα, στο πρώτο μέρος της πτυχιακής εργασίας περιγράφεται αναλυτικά το βασικό σύστημα μετάδοσης για εκπομπή σήματος επίγειας ψηφιακής τηλεόρασης. Στη συνέχεια, πραγματοποιείται η υλοποίηση της μεθόδου σε γλώσσα προγραμματισμού C.

Τέλος, η παρουσίαση ολοκληρώνεται με την εμφάνιση των αποτελεσμάτων της προσομοίωσης.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1^ο	
ΕΙΣΑΓΩΓΗ	7
1.1 Εισαγωγή στο πρότυπο DVB	7
1.2 Το πρότυπο της επίγειας ψηφιακής τηλεόρασης DVB-T	8
1.3 Ανάλυση του συστήματος μετάδοσης DVB-T.....	9
ΚΕΦΑΛΑΙΟ 2^ο	
ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΜΕΘΟΔΟΥ	11
2.1 Προσαρμογή ροής μεταφοράς και τυχαιοποίηση για διασπορά ενέργειας	11
2.2 Εξωτερική κωδικοποίηση	12
2.3 Εξωτερική αναδιάταξη (interleaving) με κώδικα Reed-Solomon	13
2.4 Εσωτερική κωδικοποίηση με διατρητό συνελκτικό κώδικα	15
2.5 Εσωτερική αναδιάταξη	16
2.6 Εισαγωγή Guard Interval	21
2.7 Διαμόρφωση QAM και απεικόνιση	22
2.8.1 Αρχή Λειτουργίας OFDM	26
2.8.2 Δομή ενός OFDM πλαισίου.....	28
2.8.3 Σήματα με πληροφορίες αναφοράς	28
2.9 Μετατροπéας Ψηφιακού σε Αναλογικό (DAC).....	30
ΚΕΦΑΛΑΙΟ 3^ο	
ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΜΕΘΟΔΟΥ ΣΕ ΓΛΩΣΣΑ C	31
3.1 Επεξήγηση κώδικα αρχείου“COFDM-Rayleigh.cpp“.....	31
3.2 Πηγαίος κώδικας MATLAB – Αρχείο “OFDM_Noise_k .m”.....	35
3.3 Πηγαίος κώδικας MATLAB – Αρχείο “my_qammod.m”.....	37

3.4 Πηγαίος κώδικας MATLAB –Αρχείο “ my_qamdemod.m”	38
3.5 Πηγαίος κώδικας σε C.....	39
ΚΕΦΑΛΑΙΟ 4^ο	
ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΗΣ.....	48
Παράρτημα Α: Παράμετροι OFDM	51
Παράρτημα Β: Μορφή μετάδοσης TPS.....	52
Παράρτημα Γ: Τεχνικά χαρακτηριστικά DVB-T.....	54
Συνομογραφίες.....	56
Βιβλιογραφία.....	57

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αντίρριο, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Ονοματεπώνυμο, Υπογραφή

2. Ονοματεπώνυμο, Υπογραφή

3. Ονοματεπώνυμο, Υπογραφή

ΚΕΦΑΛΑΙΟ 1^ο

ΕΙΣΑΓΩΓΗ

1.1 Εισαγωγή στο πρότυπο DVB

Το DVB ή Digital Video Broadcasting αναφέρεται στην μετάδοση ψηφιακού βίντεο το οποίο είναι ένα απαραίτητο στοιχείο για παγκόσμια μετάδοση και αντιπροσωπεύει ένα σετ από ψηφιακά τηλεοπτικά στάνταρντ. Όλα τα DVB standards υποστηρίζουν το MPEG-2 για συμπίεση εικόνας και ήχου. Όπως και για την πολυπλεξία. Αυτά τα στάνταρντ παρέχουν υποστήριξη για διάφορες προσαρμογές της DVB τεχνολογίας συμπεριλαμβάνοντας:

1. **Τα δορυφορικά συστήματα DVB-S και DVB-S2** Το σύστημα DVB-S έχει σχεδιαστεί για την εκπομπή ψηφιακής τηλεόρασης στην περιοχή συχνοτήτων 10,7 ως 12,75 GHz. Χρησιμοποιείται η διαμόρφωση (Quaternary Phase Shift Keying –QPSK) κωδικοποιημένη κατά τον κώδικα του Gray, η οποία δίνει καλύτερη χρήση του φάσματος συχνοτήτων και ο ρυθμός μεταφοράς που επιτυγχάνεται είναι 54Mbps Το σύστημα προβλέπεται να εκπέμπει απευθείας στους καταναλωτές, Υπάρχουν δύο τύποι ψηφιακού σήματος για τη δορυφορική τηλεόραση:
 - Ελεύθερο: Αυτός τύπος σήματος είναι διαθέσιμος παγκοσμίως
 - Συνδρομητικό : Σε αυτό τον τύπο τα τηλεοπτικά σήματα είναι κρυπτογραφημένα

Οι συνδρομητές παραλαμβάνουν έναν αποκωδικοποιητή προγράμματος, ο οποίος αποκρυπτογραφεί τα σήματα των κρυπτογραφημένων προγραμμάτων.

Το σύστημα DVB-S2 έχει καλύτερη απόδοση περίπου κατά 30% αυτό οφείλεται χάρη στη συμπίεση που επιτυγχάνεται με βάση τον αλγόριθμο MPEG-4. Τέλος, υποστηρίζονται 4 διαμορφώσεις 8-PSK, 16-APSK, 32-APSK.

2. **Το καλωδιακό σύστημα DVB-C** Αυτό το σύστημα μεταδίδει συμπιεσμένο βίντεο και ήχο σύμφωνα με τον αλγόριθμο MPEG-2 μέσω καλωδίου και διαμορφώνεται με βάση την QAM. Ο ρυθμός μεταφοράς που επιτυγχάνεται είναι 38,5Mbps.
3. **Το σύστημα επίγειας ψηφιακής μετάδοσης DVB-T** Αυτό το σύστημα λειτουργεί στον ευρωπαϊκό χώρο και περιορίζεται αποκλειστικά στην περιγραφή των λειτουργιών του διαμορφωτή. Αυτός με τη σειρά του, δέχεται από τον πολυπλέκτη το ρεύμα μεταφοράς MPEG-2 που περιέχει πολυπλεγμένες τις υπηρεσίες εικόνας, ήχου και δεδομένων υπό μορφή σήματος βασικής ζώνης και παράγει το προς μετάδοση RF (Radio Frequency) σήμα. Η λήψη πραγματοποιείται με μια απλή κεραία. Παρόμοια σύστημα χρησιμοποιούνται στην Β.Αμερική (ATSC) και στην Ιαπωνία (ISDB-T).
4. **Το σύστημα επίγειας ψηφιακής μετάδοσης για φορητούς υπολογιστές DVB-H.** Οι προδιαγραφές του DVB-H περιγράφουν τον τρόπο με τον οποίο οι φορητές συσκευές, όπως τα κινητά τηλέφωνα, μπορούν να λάβουν και να αποκωδικοποιήσουν ψηφιακό τηλεοπτικό σήμα. Η τεχνολογία DVB-H είναι συμβατή με το σύστημα DVB-T και λαμβάνονται υπόψη στις ειδικές ανάγκες των φορητών ψηφιακών συσκευών. (μέγεθος, βάρος, εξοικονόμηση ενέργειας). Η ψηφιακή εκπομπή εικόνας και ήχου είναι ικανή να προσφέρει μεγαλύτερες αναλύσεις εικόνας καθώς και πολυκάναλο ήχο. Αυτό είναι εφικτό με τους αλγόριθμους MPEG οι οποίοι συμπιέζουν αρκετά τα ψηφιακά βίντεο ώστε να απαιτείται

μικρό εύρος ζώνης για την εκπομπή τους, χωρίς να αλλοιωθεί η ποιότητα της εικόνας και του ήχου.

1.2 Το πρότυπο της επίγειας ψηφιακής τηλεόρασης DVB-T

Το DVB-T (Digital Video Broadcasting for Terrestrial) είναι το ευρωπαϊκό στάνταρτ για την εκπομπή επίγειας ψηφιακής τηλεόρασης περιορίζεται αυστηρά στην περιγραφή των λειτουργιών του διαμορφωτή. Αναπτύχθηκε από το DVB Project και εγκρίθηκε από το Ευρωπαϊκό Ινστιτούτο Τηλεπικοινωνιακών Συστημάτων ETSI (European Telecommunications Standards Institute) για πρώτη φορά το 1997. Η πρώτη εκπομπή ψηφιακού τηλεοπτικού σήματος ξεκίνησε στην Σουηδία το 1998 και το Ηνωμένο Βασίλειο. Το DVB-T έχει εγκριθεί για την ψηφιακή τηλεοπτική μετάδοση και εφαρμόζεται σε πολλές χώρες σε όλο τον κόσμο, συμπεριλαμβανομένης και της Ελλάδας.

Το πρότυπο DVB-T συνδυάζει την OFDM τεχνολογία με σύνθετες τεχνικές ισοστάθμισης και κωδικοποίησης, εισάγοντας έτσι την έννοια του κωδικοποιημένου OFDM (Coded OFDM-COFDM). Το σχήμα COFDM χρησιμοποιεί ένα μεγάλο αριθμό φερόντων συχνοτήτων, κάθε ένα από αυτά μπορεί να διαμορφωθεί κατά QPSK, 16QAM ή 64QAM. Έτσι, η πληροφορία θα καταταξιωθεί ομοιόμορφα στο φάσμα και, σε συνδυασμό με κωδικοποίηση και αναδιάταξη δύο επιπέδων, το σήμα αποκτά μεγάλη ευρωστία ακόμη σε περιβάλλον με φαινόμενα πολυόδευσης (multipath) και ισχυρών διαλείψεων.

Ο διαμορφωτής δέχεται από τον πολυπλέκτη την τελική ροή μεταφοράς MPEG-2 στην οποία περιέχονται πολυπλεγμένες οι υπηρεσίες εικόνας, ήχου και δεδομένων με τη μορφή σήματος βασικής ζώνης και τέλος στην έξοδό του παράγει το σήμα RF που πρόκειται να μεταδοθεί

Άξιο αναφοράς είναι το γεγονός ότι παρουσιάζεται η δυνατότητα ιεραρχικής διαμόρφωσης δύο επιπέδων η οποία δίνει μεγαλύτερη ευελιξία στο σχεδιασμό ενός συστήματος καθώς επιτρέπει από τον ίδιο πομπό να εξυπηρετούνται ταυτόχρονα σταθερή ή και φορητή-κινητή λήψη, παρά τις διαφορετικές απαιτήσεις της κάθε λειτουργίας. Η ροή βίντεο (MPEG-TS) κωδικοποιείται σε μια υψηλής προτεραιότητας (HP-High Priority) και σε μια χαμηλής προτεραιότητας ροή (LP-Low Priority).

Το DVB-T αποτελεί το πιο πολύπλοκο ψηφιακό σύστημα ψηφιακής τηλεόρασης οι πιο σημαντικές απαιτήσεις αναφέρονται παρακάτω:

- Θα πρέπει να διαθέτει ισχυρή προστασία απέναντι στα σφάλματα όσο αναφορά την κωδικοποίηση του καναλιού και την διαμόρφωση. Έτσι εξασφαλίζεται μηδαμινός ρυθμός σφαλμάτων (με την προϋπόθεση ότι ο λόγος του σήματος προς τον θόρυβο είναι μεγαλύτερος κάποιου ορίου).
- Είναι σχεδιασμένο να λειτουργεί στα ίδια κανάλια με την συμβατική τηλεόραση δηλ στη ζώνη συχνοτήτων των UHF (470 - 862 MHz) και VHF (174 - 230 MHz) ενώ το εύρος ζώνης του εκπεμπόμενου σήματος είναι 6, 7 ή 8 MHz.
- Το σύστημα DVB-T πρέπει να έχει τη μέγιστη δυνατή ομοιότητα με τα DVB-S, DVB-C

προκειμένου να εξασφαλισθεί το ελάχιστο δυνατό κόστος για την κατασκευή των νέων δεκτών.

- Το σύστημα DVB-T θα λειτουργεί σε μονοσυχνотικό δίκτυο (SFN – Single Frequency Network) πράγμα που σημαίνει ότι οι γειτονικοί πομποί θα εκπέμπουν τα ίδια απολύτως δεδομένα. Επίσης, θα πρέπει να εξασφαλισθεί πλήρη κάλυψη με εξωτερική κεραία
- Προαιρετικά μπορεί να χρησιμοποιηθεί ιεραρχική διαμόρφωση.

Η χρήση της επίγειας ψηφιακής τηλεόρασης παρέχει σωρεία πλεονεκτημάτων και νέων δυνατοτήτων. Ανάμεσά τους ξεχωρίζουν:

- Απλοποίηση της εγκατάστασης του δέκτη αφού γίνεται χρήση της υπάρχουσας κεραίας.
- Απαιτείται λιγότερη ισχύς μετάδοσης.
- Δυνατότητα φορητής λήψης.
- Επιτρέπει την αύξηση του aspect ratio. Το τυπικό format 4:3 αλλάζει σε 16:9.
- Επιτρέπει πολλαπλά προγράμματα και υπηρεσίες πολυμέσων για κάθε κανάλι.
- Χρησιμοποιείται μονοσυχνотικό δίκτυο άρα απαιτούνται λιγότερες συχνότητες.
- Το τηλεοπτικό ψηφιακό σήμα αποκτά ευρωστία, είναι πιο ανθεκτικό σε παρεμβολές και σε φαινόμενα πολυόδευσης.

Στη συνέχεια του εγγράφου περιγράφεται αναλυτικά το βασικό σύστημα μετάδοσης για εκπομπή σήματος επίγειας ψηφιακής τηλεόρασης.

1.3 Ανάλυση του συστήματος μετάδοσης DVB-T

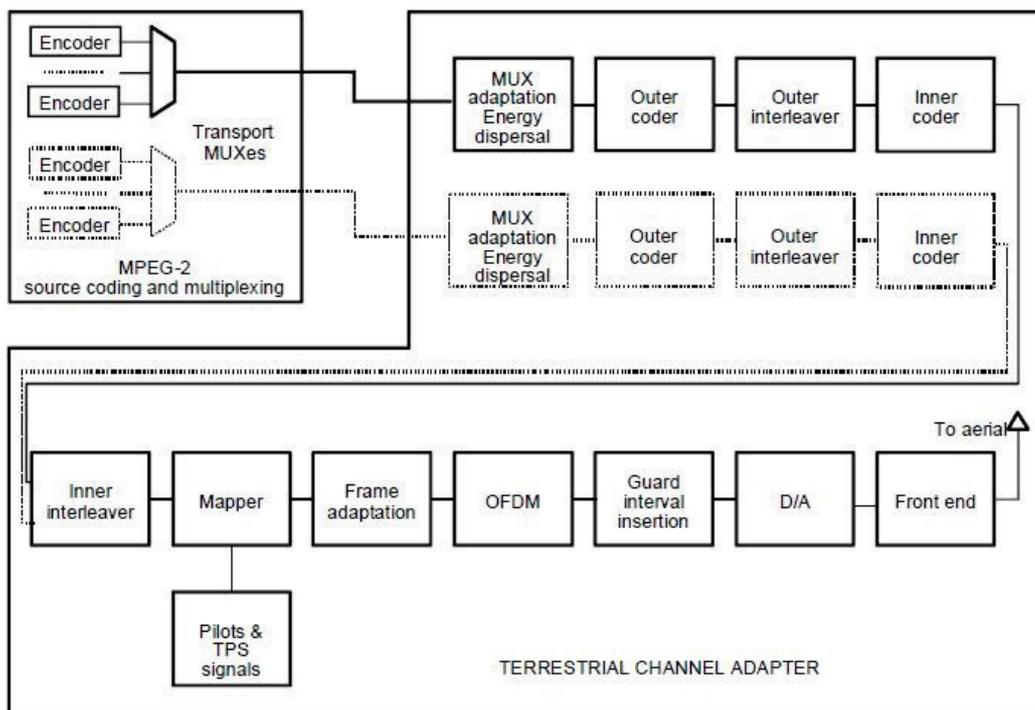
Κατά το πρότυπο ETSI EN 300 744 V1.6.1 (2009-01), το σύστημα επίγειας ψηφιακής μετάδοσης καθορίζεται από το παρακάτω σχήμα στο οποίο παρουσιάζεται το λειτουργικό μπλοκ διάγραμμα σύμφωνα με το οποίο πραγματοποιείται η προσαρμογή των τηλεοπτικού σήματος βασικής ζώνης από την έξοδο του πολυπλέκτη μεταφοράς MPEG-2 TS (Moving Pictures Experts Group Transport Stream) στα χαρακτηριστικά του επίγειου καναλιού.

Οι βασικές λειτουργίες που επιτελούνται είναι οι εξής:

- Προσαρμογή πολύπλεξης μεταφοράς (transport multiplex adaptation) και τυχαιοποίηση για διασπορά ενέργειας
- Εξωτερική κωδικοποίηση
- Εξωτερική αναδιάταξη (interleaving) με κώδικα Reed-Solomon
- Εσωτερική κωδικοποίηση με διατρητό συνελκτικό κώδικα
- Εσωτερική αναδιάταξη
- Αντιστοίχιση και διαμόρφωση φερόντων σημάτων
- Μετάδοση με τη χρήση OFDM

Τοποθεσία	Interface	Τύπος Interface	Σύνδεση
Σταθμός εκπομπής	Εισόδου	Πολύπλεξη MPEG-2 TS	Απο Πολυπλέκτη MPEG-2
	Εξόδου	Σήμα RF	Προς διεπαφή αέρα
Εγκατάσταση δέκτη	Εισόδου	Σήμα RF	Απο διεπαφή αέρα
	Εξόδου	Πολύπλεξη MPEG-2 TS	Προς Πολυπλέκτη MPEG-2

Πίνακ.1 Interfaces για baseline σύστημα μετάδοσης

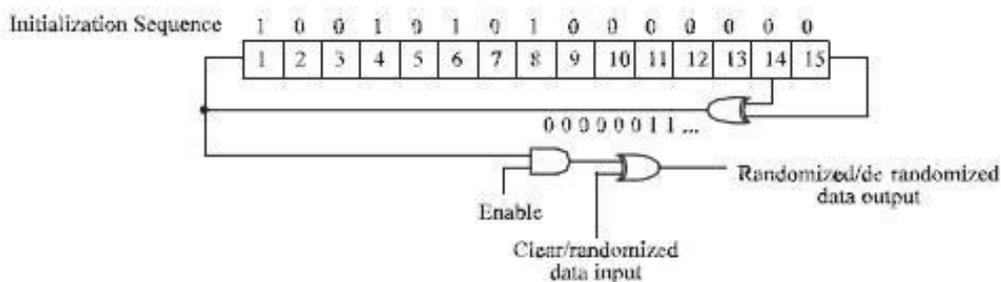


Σχ.1.3.1 Διάγραμμα ενός συστήματος εκπομπής DVB-T.

ΚΕΦΑΛΑΙΟ 2^ο ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΜΕΘΟΔΟΥ

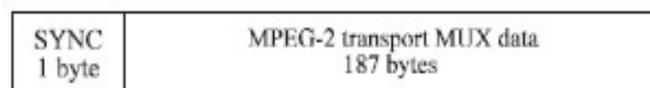
2.1 Προσαρμογή ροής μεταφοράς και τυχαιοποίηση για διασπορά ενέργειας

Η προσαρμογή ροής μεταφοράς είναι το πρώτο στάδιο επεξεργασίας του σήματος βασικής ζώνης. Η ροή MPEG-2 που προκύπτει από την έξοδο του πολυπλέκτη υφίστανται μια διαδικασία τυχαιοποίησης έτσι ώστε να εξασφαλιστεί ότι η πληροφορία θα κατανεμηθεί ομοιόμορφα σε όλο το εύρος συχνοτήτων. Αυτό επιτυγχάνεται με την εισαγωγή της σε έναν περιπλέκτη (scrambler) ο οποίος χρησιμοποιεί μία γεννήτρια ψευδοτυχαίας δυαδικής ακολουθίας (Pseudo Random Binary Sequence – PRBS generator). Επιπρόσθετα, εξασφαλίζεται ότι θα υπάρχουν αρκετές δυαδικές μεταβολές βοηθώντας στην αντιμετώπιση λαθών κατά τη διάρκεια της μετάδοσης.



Σχ. 2.1.1 Τυχαιοποίηση δεδομένων ροής μεταφοράς Mpeg-2

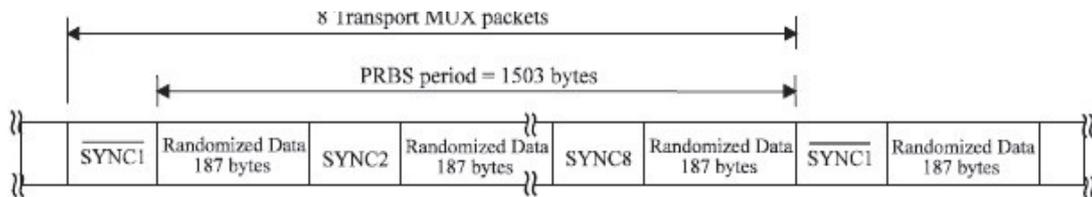
Η ροή μεταφοράς MPEG-2 που προκύπτει από την έξοδο του πολυπλέκτη οργανώνεται σε πακέτα προκαθορισμένου μήκους της τάξεως των **188** bytes και στα οποία συμπεριλαμβάνεται ένα byte συγχρονισμού.



Σχ.2.2.2: Μορφή πλαισίου μήκους 188 bytes

Τα τυχαιοποιημένα δεδομένα προκύπτουν με εφαρμογή XOR μεταξύ του δεδομένου και της εξόδου της ακολουθίας. Η αρχική ακολουθία που δημιουργείται στους 15 καταχωρητές ορίζεται ως «100101010000000». Η γεννήτρια ψευδοτυχαίας ακολουθίας αρχικοποιείται κάθε 8 πακέτα μεταφοράς, χωρίς να επηρεαστεί το byte συγχρονισμού. Για να παρέχουμε ένα σήμα αρχικοποίησης για τον descrambler (διαδικασία «αποτυχοποίησης») το byte συγχρονισμού του

πρώτου πακέτου σε ένα γκρουπ από 8 πακέτα τίθεται ίσο με το συμπληρωματικό του μη τυχαιοποιημένο. Αυτή η διαδικασία λέγεται transport multiplex adaptation και παριστάνεται ως εξής:



Σχ. 2.2.3 Διάγραμμα της προσαρμογής πολύπλεξης μεταφοράς

Πιο αναλυτικά, το πρώτο bit που παράγεται από την ακολουθία εφαρμόζεται στο πρώτο bit του πρώτου byte που ακολουθεί το συμπληρωματικό byte συγχρονισμού. Για να διευκολυνθούν οι υπόλοιπες διαδικασίες συγχρονισμού η ακολουθία παράγει τιμές αλλά η έξοδος της δεν θα είναι ενεργοποιημένη. Οπότε όλα τα επόμενα bytes συγχρονισμού των 7 πακέτων που θα ακολουθήσουν παραμένουν μη τυχαιοποιημένα. Η περίοδος της ακολουθίας PRBS θα είναι $188-1 = 1503$ bytes. Τέλος, η διαδικασία τυχαιοποίησης θα είναι ενεργή σε δυο περιπτώσεις: όταν δεν υπάρχει είσοδος bitsream στο διαμορφωτή ή όταν αυτή δεν είναι συμβατή με πρότυπο μεταφοράς MPEG2TS (1 byte συγχρονισμού + 187 πακέτα του 1 byte).

2.2 Εξωτερική κωδικοποίηση

Πρόκειται για το πρώτο στάδιο διόρθωσης λαθών το οποίο εφαρμόζεται στα μεταδιδόμενα δεδομένα χρησιμοποιώντας έναν μη δυαδικό κώδικα, τον αλγόριθμο Reed-Solomon, ο οποίος επιτρέπει τη διόρθωση μέχρι και 8 λανθασμένων bytes για κάθε πακέτο των 188 bytes.

Ένας κώδικας Reed-Solomon προσδιορίζεται ως $RS(n,k)$ με σύμβολα s -bit. Ο κωδικοποιητής παίρνει k σύμβολα δεδομένων των s bits το καθένα και προσθέτει σύμβολα ισοτιμίας για να δημιουργήσει μια κωδική λέξη των n συμβόλων. Υπάρχουν $n-k$ σύμβολα ισοτιμίας των s bits το καθένα. Ένας αποκωδικοποιητής Reed-Solomon μπορεί να διορθώσει μέχρι t σύμβολα που περιέχουν σφάλματα, όπου $2t = n-k$. το μέγιστο μήκος της κωδικής λέξης $n = 2^s - 1$.

Η εξωτερική κωδικοποίηση πραγματοποιείται στη δομή του πακέτου εισόδου. Ο κώδικας Reed-Solomon **RS(204,188, t=8)** υλοποιείται προσθέτοντας 51 bytes (όλα μηδέν) πριν από τα bytes της πληροφορίας στην είσοδο του κωδικοποιητή RS(255,239,8) στη συνέχεια τα μηδενικά bytes απορρίπτονται και η τελική κωδική λέξη n αποτελείται από 204 bytes. Ο συντομευμένος κώδικας Reed-Solomon εφαρμόζεται σε κάθε τυχαιοποιημένο πακέτο ώστε να

παραχθεί ένα πακέτο το οποίο είναι προστατευμένο από σφάλματα . Ο κώδικας Reed Solomon επιτρέπει τη διόρθωση μέχρι και 8 λανθασμένων bytes. Η κωδικοποίηση εφαρμόζεται και στο byte συγχρονισμού πακέτου, είτε του κανονικού είτε του συμπληρωματικού.

Πολυώνυμο γεννήτρια κώδικα: $g(x) = (x+\lambda^0)(x+\lambda^1)(x+\lambda^2)...(x+\lambda^{15})$

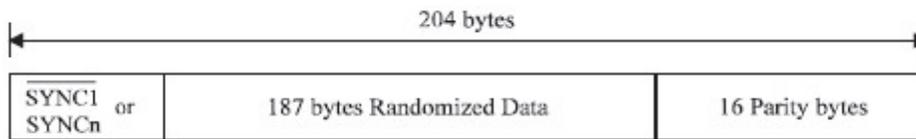
Πολυώνυμο γεννήτρια πεδίου: $p(x) = x^8 + x^4 + x^3 + x^2 + 1$

Ένας αποκωδικοποιητής είναι ικανός να διορθώσει μέχρι t σφάλματα ή να ανιχνεύσει μέχρι $2t$ σφάλματα. Η πληροφορία ανίχνευσης συχνά να προέρχεται από τον αποδιαμορφωτή σε ένα ψηφιακό τηλεπικοινωνιακό σύστημα (π.χ. ο αποδιαμορφωτής σημειώνει σύμβολα που είναι πιθανό να περιέχουν σφάλματα).

Υπάρχουν 3 πιθανές περιπτώσεις κατά την αποκωδικοποίηση μιας κωδικής λέξης:

- i. Αν $2s + r < 2t$ (όπου s σφάλματα, r ανιχνεύσεις), τότε η αρχική μεταδιδόμενη κωδική λέξη είναι δυνατόν να ανακτηθεί.
- ii. Ο αποκωδικοποιητής θα ανιχνεύσει ότι δεν μπορεί να ανακτήσει την αρχικό κωδική λέξη.
- iii. Δε θα γίνει σωστά η αποκωδικοποίηση της κωδικής λέξης και θα ανακτηθεί μια λανθασμένη.

Το πλεονέκτημα της χρήσης κώδικα Reed-Solomon είναι ότι η πιθανότητα ενός λάθους στα κωδικοποιημένα δεδομένα είναι πολύ χαμηλότερη από την πιθανότητα λάθους αν δε χρησιμοποιηθεί αυτή.

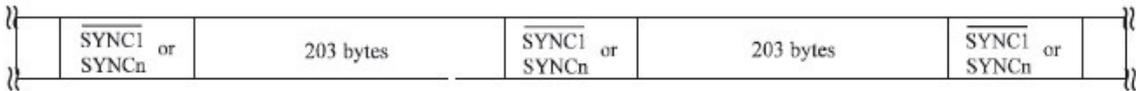


Σχ. 2.2.4 Πακέτο κωδικοποιημένο με RS(204,188)

2.3 Εξωτερική αναδιάταξη

Η συνελκτική αναδιάταξη χρησιμοποιείται για να αναδιατάξει την μεταδιδόμενη ακολουθία δεδομένων έτσι ώστε να γίνει πιο ανθεκτική σε μεγάλες ακολουθίες σφαλμάτων. Η διαδικασία περιγράφεται παρακάτω:

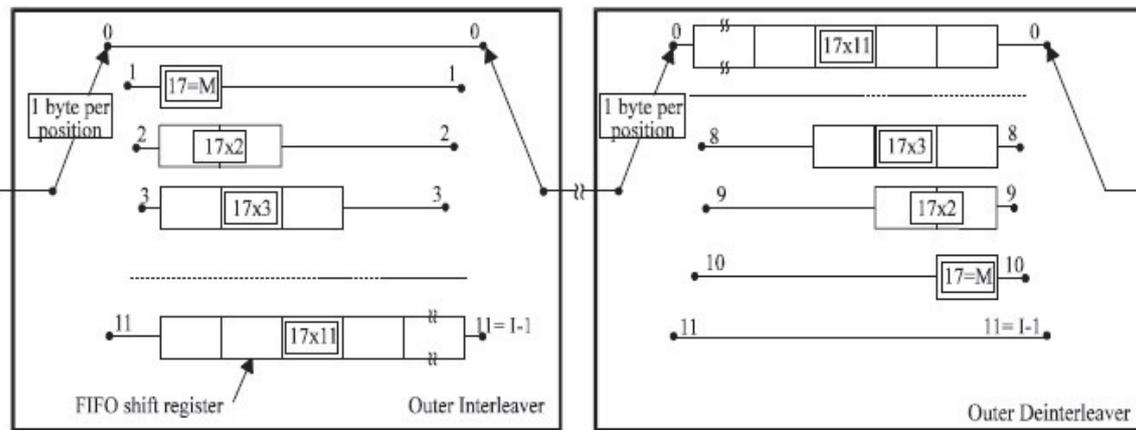
Σε αυτό το στάδιο αν υπάρξουν λανθασμένα bytes τότε είναι πιθανό αυτά τα bytes να μην μπορούν να διορθωθούν από τον κώδικα Reed-Solomon. Σε αυτή την περίπτωση, θα γίνει αναδιάταξη των bytes ώστε τα γειτονικά bytes να ανήκουν σε διαφορετικά πακέτα κατά την μετάδοση. Στη συνέχεια, τα λανθασμένα bytes θα κατανεμηθούν ώστε να είναι πιο εύκολη η διόρθωσή τους. Τα αναδιεταγμένα bytes αποτελούνται από τα πακέτα τα οποία είναι προστατευμένα από σφάλμα και οριοθετούνται από τα bytes συγχρονισμού MPEG-2 (συμπληρωματικά ή μη), διατηρώντας την περιοδικότητα των 204 bytes.



Σχ. 2.3.1 Μορφή πλαισίων μετά την συνελκτική αναδιάταξη

Ακολουθεί η διαδικασία της συνελκτικής αναδιάταξης η οποία περιγράφεται ως εξής:

Ο διεμπλοκείας αποτελείται από $I=12$ κλάδους οι οποίοι είναι κυκλικά συνδεδεμένοι στην είσοδο της ροής των bytes στον διακόπτη εισόδου. Κάθε κλάδος j αποτελεί έναν FIFO (First-In, First-Out) καταχωρητή ολίσθησης (shift register) με βάθος $j \times M$ κελιά όπου $N=204$, $j=0, \dots, 11$, $M=N/I$ δηλ. 17 κελιά. Τα κελιά του FIFO περιέχουν 1 byte, επίσης οι διακόπτες εισόδου και εξόδου θα είναι συγχρονισμένοι. Για λόγους συγχρονισμού τα SYNC bytes θα δρομολογούνται πάντα από τον κλάδο "0" του διαπλοκείας (interleaver).



Σχ.2.3.2 Διάγραμμα διαπλοκείας/αποπλέκτη

Παρατηρώντας το παραπάνω διάγραμμα, το 1^ο byte μεταδίδεται χωρίς καθυστέρηση. Το 2^ο byte με καθυστέρηση 17bytes κ.ο.κ. μέχρι το 12^ο που θα μεταδοθεί με καθυστέρηση $17 \times 11 = 187$ bytes. Στη λήψη οι καθυστερήσεις αντιστρέφονται έτσι ώστε το 1^ο byte να καθυστερήσει κατά $17 \times 11 = 187$ bytes, το 2^ο κατά $17 \times 10 = 170$ bytes κ.ο.κ μέχρι το 12^ο που θα ληφθεί χωρίς καθυστέρηση. Με αυτή τη μέθοδο όλα τα bytes θα έχουν υποστεί καθυστέρηση στη λήψη ίση $17 \times 11 = 187$ bytes.

Η εξωτερική αναδιάταξη πραγματοποιείται για την αντιμετώπιση σφαλμάτων μεγάλης διάρκειας. Τα σφάλματα δεν εμφανίζονται ανεξάρτητα μεταξύ τους, αλλά ούτε και σαν καθορισμένοι καταγισμοί. Χάρη στην καθυστέρηση που εισέρχεται είναι πλέον αντιληπτό, ότι ο καταγισμός σφαλμάτων μήκους 12 ψηφίων ή λιγότερου, θα επηρεάσει όχι περισσότερο από ένα ψηφίο σε κάθε σειρά.

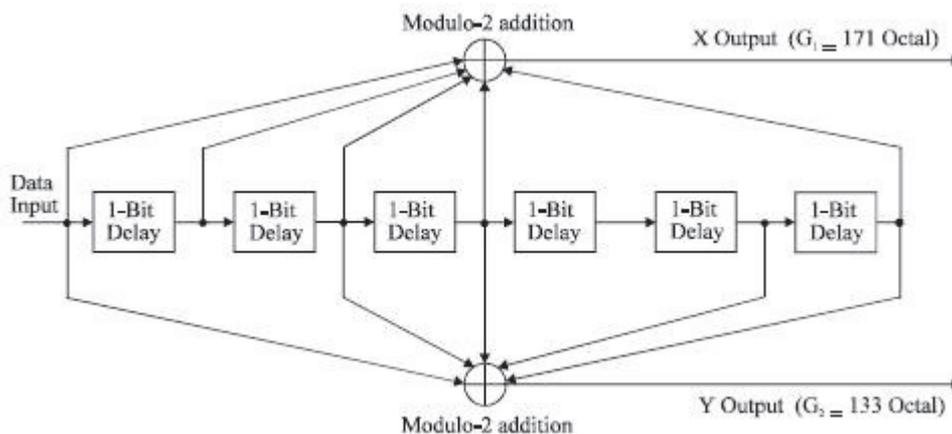
2.4 Εσωτερική κωδικοποίηση με διατρητό συνελκτικό κώδικα

Το δεύτερο επίπεδο διόρθωσης λαθών επιτυγχάνεται μέσω της τεχνικής συνελκτικής κωδικοποίησης (Convolutional Coding). Στην ουσία στοχεύουμε να αυξήσουμε την απόδοση της κωδικοποίησης Reed-Solomon. Το πρότυπο είναι ικανό να υποστηρίξει 5 ρυθμούς κωδικοποίησης. Από τον ισχυρότερο προς τον ασθενέστερο αυτοί είναι 1/2, 2/3, 3/4, 5/6 και 7/8. Ο ρυθμός κωδικοποίησης καθορίζει τον αριθμό των bits στην είσοδο προς τον αριθμό των bits που προκύπτουν στην έξοδο.

Ο λόγος που απαιτείται αποδοτικότερη προστασία έναντι στα σφάλματα οφείλεται στο γεγονός ότι η ποιότητα των σημάτων στα κανάλια μετάδοσης μεταβάλλεται και είναι πιθανόν ένας μεγάλος αριθμός από δυαδικά ψηφία να μην μπορούν να διορθωθούν με την κωδικοποίηση Reed-Solomon. Αυτό έχει ως αποτέλεσμα να αλλοιώνεται το σήμα όπως για παράδειγμα να υπάρξει ένα χτύπημα κερανού ή από διάφορες γειτονικές παρεμβολές από ηλεκτρικές συσκευές. Η προστασία του κάθε πακέτου μετάδοσης από τέτοιου είδους αλλοιώσεις είναι μη οικονομική, εφόσον αυτές συμβαίνουν σπάνια.

Σε αυτό το στάδιο εφαρμόζεται η συνελκτική κωδικοποίηση, κατά την οποία τα δεδομένα αφού κωδικοποιηθούν με τη μέθοδο Forward Error Correction - FEC, θα τροφοδοτηθούν σε μια μνήμη RAM και στη συνέχεια θα μεταδοθούν αναδιαταγμένα. Χρησιμοποιώντας μια δεύτερη μνήμη RAM κατά τη λήψη, τα δεδομένα τοποθετούνται στην αρχική τους δομή. Το αποτέλεσμα της κωδικοποίησης είναι να συναρμολογηθούν τα δυαδικά ψηφία που έχουν υποστεί σφάλμα και να μετατραπούν σε έναν μεγάλο αριθμό ενιαίων εσφαλμένων συμβόλων, τα οποία όμως θα είναι εύκολο να διορθωθούν.

Το σύστημα DVB-T επιτρέπει ένα εύρος διατρητικών συνελκτικών κωδικών, οι οποίοι βασίζονται σε έναν μητρικό συνελκτικό κώδικα με ρυθμό 1/2 και 64 καταστάσεις. Έτσι θα μπορέσει να γίνει η επιλογή του καταλληλότερου ρυθμού διόρθωσης σφαλμάτων για μια υπηρεσία ή ένα ρυθμό δεδομένων. Αυτό ισχύει στην ιεραρχική και στη μη ιεραρχική κατάσταση μετάδοσης. Οι γεννήτριες πολυώνυμα του μητρικού κώδικα είναι $G_1 = 171$ Oct και $G_2 = 133$ Oct για τις εξόδους X,Y.



Σχ.2.4.1 Μητρικός Συνελκτικός Κώδικας με Ρυθμό Κώδικα 1/2

Τα bits εισάγονται στην γεννήτρια και υφίστανται καθυστέρηση ανάλογα με τους καταχωρητές ολίσθησης. Αν χρησιμοποιηθεί ιεραρχική μετάδοση δύο επιπέδων, κάθε ένας από τους δυο παράλληλους κωδικοποιητές καναλιού μπορεί να έχει το δικό του ρυθμό κωδικοποίησης. Η μορφή της διάτρησης μπορεί να αλλάξει ανάλογα με το ρυθμό κωδικοποίησης αφού από μια σειρά εξόδων x και y επιλέγεται κάθε φορά μόνο ένα υποσύνολο τους για μετάδοση. Αυτό συνεπάγεται την αύξηση του ρυθμού κώδικα. Στον παρακάτω πίνακα ορίζεται η διάτρηση για τους απαιτούμενους ρυθμούς κώδικα:

Code Rates r	Puncturing pattern	Transmitted sequence (after parallel-to-serial conversion)
1/2	X: 1 Y: 1	$X_1 Y_1$
2/3	X: 10 Y: 11	$X_1 Y_1 Y_2$
3/4	X: 101 Y: 110	$X_1 Y_1 Y_2 X_3$
5/6	X: 10101 Y: 11010	$X_1 Y_1 Y_2 X_3 Y_4 X_5$
7/8	X: 1000101 Y: 1111010	$X_1 Y_1 Y_2 Y_3 Y_4 X_5 Y_6 X_7$

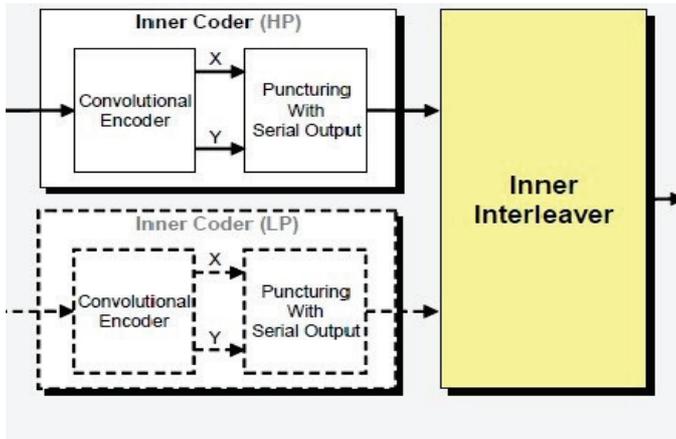
Πίνακ. 2 Σχήμα διάτρησης για την εκπεμπόμενη ακολουθία

Η ακριβής θέση των ψηφίων αυτών (με την αντίστοιχη επαναληψιμότητα) δίδεται από την μορφή διάτρησης με μορφή μάσκας (1: το ψηφίο αποστέλλεται, 0: το ψηφίο αφαιρείται). Η τρίτη στήλη απεικονίζει τη μορφή η οποία θα έχει η ακολουθία των ψηφίων που μεταδίδονται μετά την εκάστοτε διάτρηση. Είναι χρήσιμο να αναφέρουμε ότι το X_1 αποστέλλεται πάντα πρώτο, στην έναρξη κάθε πακέτου, το MSB (Most Significant Bit) του byte συγχρονισμού βρίσκεται στην είσοδο data input και ότι το πρώτο συνελκτικά κωδικοποιημένο bit ενός συμβόλου είναι πάντα το X_1 .

2.5 Εσωτερική αναδιάταξη

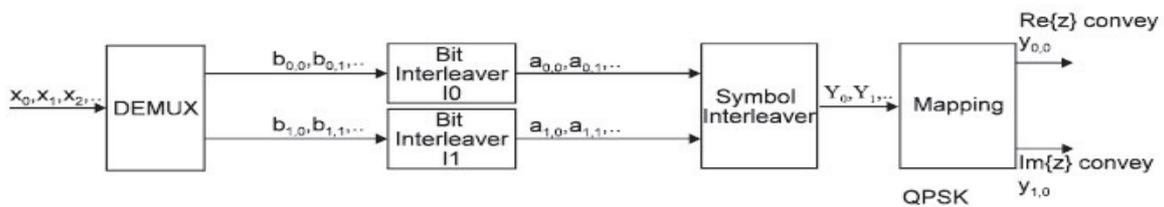
Στον εσωτερικό διεπλοκέα (interleaver), οι ακολουθίες των δεδομένων αναδιατάσσονται ξανά με μοναδικό σκοπό να μειωθεί η επιρροή των σφαλμάτων. Αυτή η διαδικασία περιλαμβάνει αναδιάταξη σε επίπεδο bits και σε επίπεδο συμβόλων.

Αναδιάταξη σε επίπεδο bit



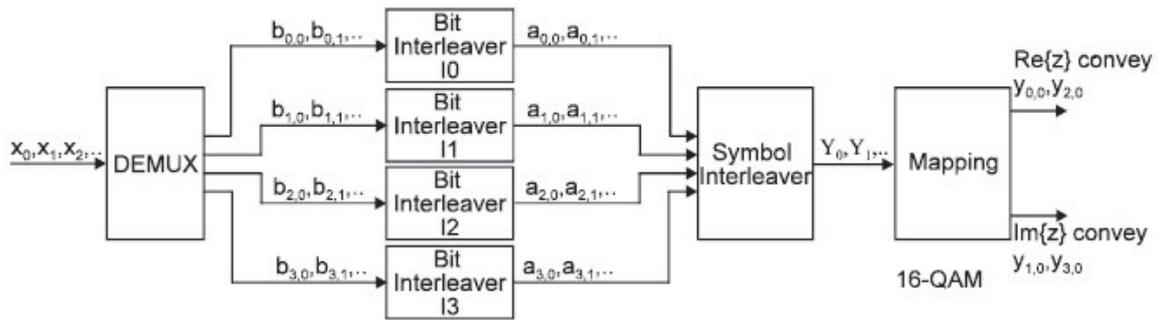
Σχ. 2.5.1 Εσωτερική Κωδικοποίηση και Εσωτερική αναδιάταξη

Η είσοδος, αποτελείται μέχρι και από δύο ροές bit, αποπλέκεται σε v υπό-ροές, όπου $v=2$ για QPSK, $v=4$ για 16-QAM και $v=6$ για 64-QAM. Στην μη ιεραρχική λειτουργία, η μονή ροή εισόδου αποπλέκεται σε v υπό-ροές. Στην ιεραρχική λειτουργία, η ροή υψηλής προτεραιότητας, αποπλέκεται σε δύο υπό-ροές και η χαμηλής προτεραιότητας ροή αποπλέκεται σε $v-2$ υπό-ροές. Αυτό εφαρμόζεται τόσο στις ομοιόμορφες όσο και στις μη ομοιόμορφες QAM.



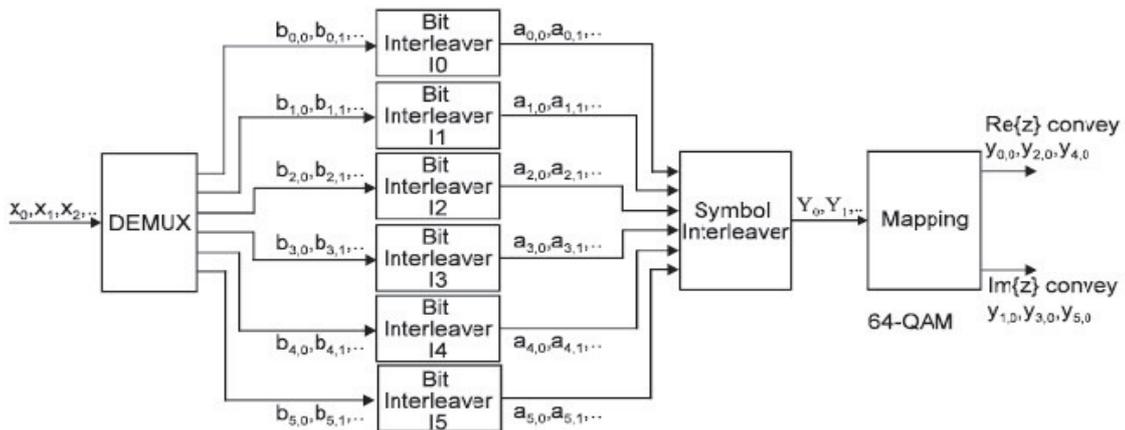
Σχ. 2.5.2 Μη ιεραρχική αναδιάταξη για QPSK ($v=2$)

Στην μη ιεραρχική λειτουργία η ροή εισόδου αποπλέκεται σε v ροές εξόδου ανάλογα με τη διαμόρφωση που χρησιμοποιείται. Για τη διαμόρφωση QPSK υπάρχουν 2 ροές εξόδου. Υπάρχουν 2 διαπλοκείς (interleavers) bit I_0 και I_1 .



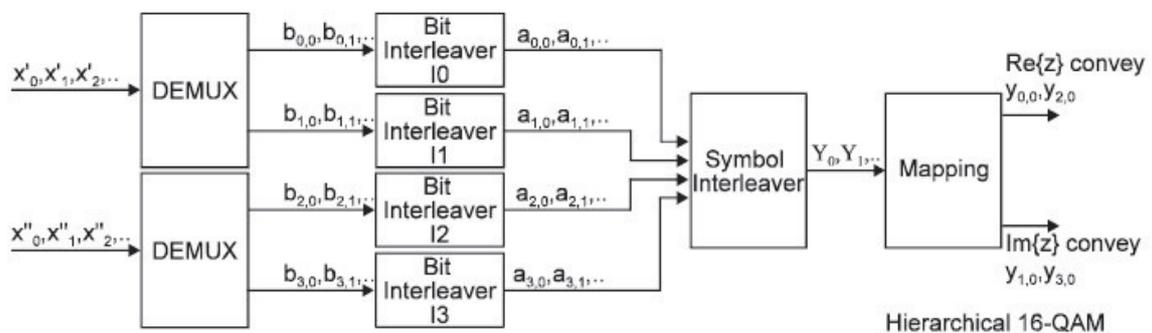
Σχ. 2.5.3 Μη ιεραρχική αναδιάταξη για 16-QAM ($v=4$)

Για διαμόρφωση 16-QAM οι ροές εξόδου είναι 2. Υπάρχουν 4 διαπλοκές (interleavers) bit I_0, I_1, I_2 και I_3 .



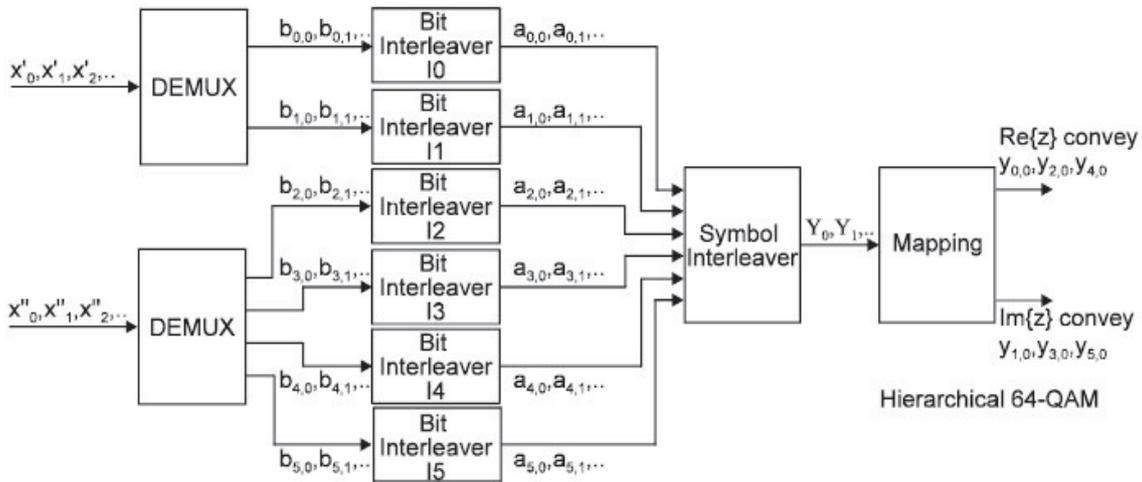
Σχ. 2.5.4 Μη ιεραρχική αναδιάταξη για 64-QAM ($v=6$)

Για διαμόρφωση 64-QAM οι ροές εξόδου είναι 4. Διαπιστώνουμε από το παραπάνω σχήμα ότι υπάρχουν 6 διαπλοκές (interleavers) bit $I_0, I_1, I_2, I_3, I_4, I_5$.



Σχ. 2.5.6 Ιεραρχική αναδιάταξη για 16-QAM ($v=4$)

Στην ιεραρχική λειτουργία, η ροή υψηλής προτεραιότητας, αποπλέκεται σε δύο υπό-ροές και η χαμηλής προτεραιότητας ροή αποπλέκεται σε $v-2$ υπό-ροές. Ανάλογα με την χρησιμοποιούμενη διαμόρφωση (16-QAM ή 64-QAM) έχουμε τα παρακάτω λειτουργικά διαγράμματα (το τονούμενο ρεύμα x' είναι το ρεύμα υψηλής προτεραιότητας ενώ το διστονούμενο το χαμηλής προτεραιότητας). Στη συνέχεια ακολουθεί η αναδιάταξη για διαμόρφωση 16-QAM. Υπάρχουν 2 διαπλοκείς (interleavers) bit I_0, I_1 , για τη ροή υψηλής προτεραιότητας και 2 διαπλοκείς για την χαμηλή ροή I_2, I_3 :



Σχ. 2.5.7 Ιεραρχική αναδιάταξη για 64-QAM ($v=2$)

Στην αναδιάταξη για διαμόρφωση 64-QAM υπάρχουν 2 διαπλοκείς (interleavers) bit I_0, I_1 και 4 διαπλοκείς I_2, I_3, I_4, I_5 για τη ροή υψηλής και για τη ροή χαμηλής προτεραιότητας αντίστοιχα. Η απόπλεξη ορίζεται ως μια απεικόνιση των bit εισόδου x_{di} στα bit εξόδου $b_{e,do}$ ως εξής:

Μη Ιεραρχικά: $x_{di} = b_{[di(mod)v](div)(v/2)+2[di(mod)(v/2)],di(div)v}$

Ιεραρχικά: $x'_{di} = b_{di(mod)2,di(div)2}$

$$x''_{di} = b_{[di(mod)(v-2)](div)((v-2)/2)+2[di(mod)((v-2)/2)]+2,di(div)(v-2)}$$

Όπου:

x_{di} ,είναι η είσοδος του αποκωδικοποιητή σε μη ιεραρχική λειτουργία

x'_{di} ,είναι η είσοδος υψηλής προτεραιότητας στον αποκωδικοποιητή

x''_{di} ,είναι η είσοδος χαμηλής προτεραιότητας στον αποκωδικοποιητή σε ιεραρχική λειτουργία

di ,είναι ο αριθμός του bit εισόδου

$b_{e,do}$,είναι η έξοδος από τον αποπλέκτη

e ,είναι ο αριθμός του αποπλεγμένου ρεύματος bit ($0 < e < v$)

do ,είναι ο αριθμός bit ενός δεδομένου ρεύματος στην έξοδο του αποπλέκτη

mod, είναι ο τελεστής modulo
 div ,είναι ο τελεστής ακέραιας διαίρεσης

Τα αποτελέσματα της απόπλεξης είναι:

QPSK	16-QAM ιεραρχική	64-QAM ιεραρχική
Το x_0 απεικονίζεται στο $b_{0,0}$	Το x'_0 απεικονίζεται στο $b_{0,0}$	Το x'_0 απεικονίζεται στο $b_{0,0}$
Το x_1 απεικονίζεται στο $b_{1,0}$	Το x'_1 απεικονίζεται στο $b_{1,0}$	Το x'_1 απεικονίζεται στο $b_{1,0}$
	Το x''_0 απεικονίζεται στο $b_{2,0}$	Το x''_0 απεικονίζεται στο $b_{2,0}$
	Το x''_1 απεικονίζεται στο $b_{3,0}$	Το x''_1 απεικονίζεται στο $b_{3,0}$
		Το x''_2 απεικονίζεται στο $b_{4,0}$
		Το x''_3 απεικονίζεται στο $b_{5,0}$

16-QAM ιεραρχική	64-QAM μη ιεραρχική
Το x_0 απεικονίζεται στο $b_{0,0}$	Το x_0 απεικονίζεται στο $b_{0,0}$
Το x_1 απεικονίζεται στο $b_{2,0}$	Το x_1 απεικονίζεται στο $b_{2,0}$
Το x_2 απεικονίζεται στο $b_{1,0}$	Το x_2 απεικονίζεται στο $b_{4,0}$
Το x_3 απεικονίζεται στο $b_{3,0}$	Το x_3 απεικονίζεται στο $b_{1,0}$
	Το x_4 απεικονίζεται στο $b_{3,0}$
	Το x_5 απεικονίζεται στο $b_{5,0}$

Η αναδιάταξη bit πραγματοποιείται μόνο στα ωφέλιμα δεδομένα. Το μέγεθος του block είναι το ίδιο για κάθε διεμπλοκέα, αλλά η ακολουθία αναδιάταξης είναι σε κάθε περίπτωση διαφορετική. Το μέγεθος block είναι 126 bits. Η διαδικασία της αναδιάταξης επαναλαμβάνεται 12 φορές για κάθε OFDM σύμβολο ωφέλιμων δεδομένων στη λειτουργία 2K και 48 φορές στη λειτουργία 8K.

Για κάθε διεμπλοκέα, το διάνυσμα εισόδου καθορίζεται από $B(e) = (b_{e,0}, b_{e,1}, b_{e,2}, \dots, b_{e,125})$, όπου e κυμαίνεται από 0 σε $v-1$.

Το διαπλεγμένο διάνυσμα εξόδου $A(e) = (a_{e,0}, a_{e,1}, a_{e,2}, \dots, a_{e,125})$ ορίζεται από: $a_{e,w} = b_{e,He(w)}$ για $w = 0, 1, 2, \dots, 125$

Όπου $He(w)$ είναι μια συνάρτηση μετάθεσης που είναι διαφορετική για κάθε διεμπλοκέα και ορίζεται ως εξής για τον καθένα:

I0: $H0(w) = w$

I1: $H1(w) = (w + 63) \text{ mod } 126$

I2: $H2(w) = (w + 105) \text{ mod } 126$

$$I3: H3(w) = (w + 42) \bmod 126$$

$$I4: H4(w) = (w + 21) \bmod 126$$

$$I5: H5(w) = (w + 84) \bmod 126$$

Οι έξοδοι των v διαπλοκέων bits ομαδοποιούνται για να δημιουργήσουν τα σύμβολα ψηφιακών δεδομένων, με τρόπο τέτοιο ώστε ακριβώς ένα σύμβολο από v bits να αποτελείται από ένα ακριβώς bit από καθέναν από τους v διεμπλοκείς. Επομένως, η έξοδος από των διαπλοκέα bit είναι μια λέξη v bit y' της οποίας το πιο σημαντικό bit είναι η έξοδος του I0 κ.ο.κ: $y'_w = (a_{0,w}, a_{1,w}, \dots, a_{v-1,w})$.

Αναδιάταξη σε επίπεδο συμβόλου

Σε αυτή τη διαδικασία πραγματοποιείται η αναδιάταξη της ψηφιακής ροής των bits στις υπόφεροντες, για να αντιμετωπιστούν οι ισχυρές διαλείψεις. Ο σκοπός του διαπλοκέα συμβόλου είναι να αντιστοιγήσει λέξεις των v bits στα 1512 (λειτουργία 2K) ή 6048 (λειτουργία 8K) ενεργά φέροντα για κάθε σύμβολο OFDM. Ο διαπλοκέας συμβόλου λειτουργεί σε block των 1512 (λειτουργία 2K) ή 6048 (λειτουργία 8K) συμβόλων δεδομένων. Άρα στη λειτουργία 2K, διαβάζονται ακολουθιακά 12 ομάδες των 126 λέξεων δεδομένων σε ένα διάνυσμα $Y' = (y'_0, y'_1, y'_2, \dots, y'_{1511})$.

Ομοίως στη λειτουργία 8K, οι ομάδες είναι 48 το μήκος δεδομένων 126 λέξεις το διάνυσμα που προκύπτει ορίζεται ως $Y' = (y'_0, y'_1, y'_2, \dots, y'_{6047})$.

Το διαπλεγμένο διάνυσμα $Y = (y_0, y_1, y_2, \dots, y'_{N_{max}-1})$ ορίζεται από τις παρακάτω σχέσεις:

$$y_{H(q)} = y'_q \text{ για μονά σύμβολα για } q = 0, \dots, N_{max}-1$$

$$y_q = y'_{H(q)} \text{ για ζυγά σύμβολα για } q = 0, \dots, N_{max}-1$$

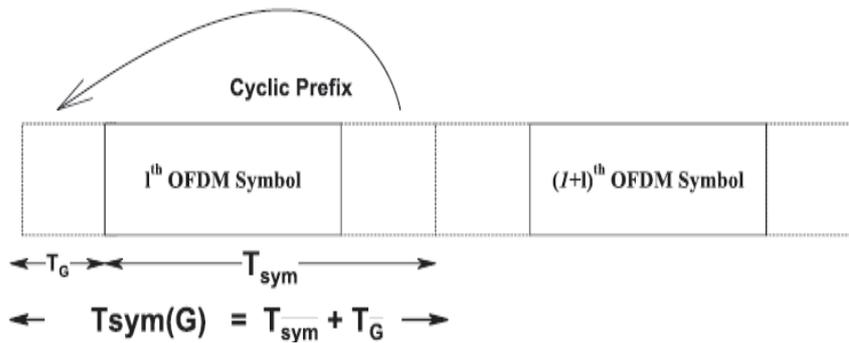
όπου $N_{max} = 1512$ στη λειτουργία 2K, $N_{max} = 6048$ στη λειτουργία 8K.

2.6 Εισαγωγή Guard Interval

Το Guard Interval είναι ένα χρονικό διάστημα το οποίο εισάγεται μεταξύ των OFDM συμβόλων ώστε να αντιμετωπιστεί το φαινόμενο της ενδοσυμβολικής παρεμβολής.

Η ύπαρξη του διαστήματος ασφαλείας παρέχει τη δυνατότητα στο δέκτη να συσχετίσει δύο ή περισσότερα όμοια σήματα που έχουν ληφθεί με ανάλογη χρονική καθυστέρηση, και παράλληλα να ενισχύσει το προς αποδιαμόρφωση σήμα. Με αυτή την ιδιότητα διευκολύνεται ο σχεδιασμός δικτύων στα οποία όλοι οι πομποί λειτουργούν στην ίδια συχνότητα (Single Frequency Network, SFN), ενώ βελτιώνει και η επίδοση του συστήματος σε περιβάλλον πολλαπλών ανακλάσεων, όπως για παράδειγμα οι αστικές περιοχές. Το μήκος αυτών των διαστημάτων ασφαλείας μπορεί να είναι $1/4, 1/8, 1/16, 1/32$.

Η διάρκεια του GI πρέπει να είναι μεγαλύτερη από διάρκεια εξάπλωσης καθυστέρησης του σήματος, ώστε να μη χαθεί η ορθογωνιότητά του και να μην εμφανιστεί παρεμβολή μεταξύ των φερόντων.



Σχ. 2.6.1 OFDM σύμβολο με κυκλικό πρόθεμα.

2.7 Διαμόρφωση QAM και απεικόνιση

Η QAM (ορθογωνική διαμόρφωση πλάτους) είναι μια μέθοδος διαμόρφωσης κατά την οποία μεταβάλλεται το πλάτος και η φάση δυο φερόντων κυμάτων. Τα φέροντα κύματα έχουν διαφορά φάσης 90° . Τα διαμορφωμένα κύματα θα αθροιστούν και θα έχουν την μορφή:

$S(t) = I(t)\cos(2\pi f_0 t) + Q(t)\sin(2\pi f_0 t)$, το φέρον $\cos(2\pi f_0 t)$ λέγεται συμφασικό (in-phase) και το $\sin(2\pi f_0 t)$ ορθογωνικό (quadrature).

Ένα φέρον είναι μια συνεχής συχνότητα που έχει τη δυνατότητα να διαμορφωθεί από ένα δεύτερο σήμα το οποίο θα μεταφέρει την πληροφορία.

Αν χρησιμοποιήσουμε μεγαλύτερο πλήθος μετατοπίσεων φάσης και πλάτους τότε επιτυγχάνουμε υψηλότερο ρυθμό μετάδοσης δεδομένων στο ίδιο εύρος ζώνης. Συνεπώς, γίνεται αποδοτικότερη χρήση του φάσματος. Ωστόσο αυξάνεται και η πιθανότητα σφάλματος λόγω θορύβου και εξασθένισης.

Οι συνδυασμοί πλάτους και φάσης στα σχήματα της QAM είναι γνωστοί ως αστερισμοί. Στόχος ενός αστερισμού είναι η αντιστοίχιση μιας ψηφιακής ακολουθίας δεδομένων σε έναν μιγαδικό αριθμό ο οποίος είναι έτοιμος προς μετάδοση.

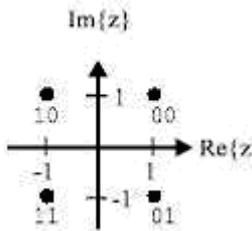
Το σύστημα χρησιμοποιεί μετάδοση πολύπλεξης ορθογώνιας διαίρεσης συχνότητας (OFDM). όλα τα φέροντα δεδομένων σε ένα πλαίσιο OFDM, θα διαμορφωθούν χρησιμοποιώντας σήματα αστερισμού QPSK, 16-QAM, 64-QAM, μη ομοιόμορφης 16-QAM ή μη ομοιόμορφης 64-QAM. Η συμμετρία για τους αστερισμούς εξαρτάται από τον παράγοντα διαμόρφωσης α , ο οποίος ορίζεται ως εξής:

Η απόσταση α ορίζεται ως ο λόγος της ελάχιστης απόστασης ανάμεσα σε δυο σημεία του αστερισμού που μεταφέρουν διαφορετικές τιμές bit υψηλής προτεραιότητας προς την ελάχιστη απόσταση δύο οποιονδήποτε σημείων του αστερισμού. Η παράμετρος α , παίρνει παρνη τις τιμές 1, 2 ή 4, αυξάνοντας παράλληλα και τα μεγέθη των τριών διαγραμμάτων.

Ο κανόνας που σχετίζει τον τύπο της διαμόρφωσης με αριθμό των bits που θα περιέχει κάθε σύμβολο είναι ο εξής: $2^m = q$, όπου m ο αριθμός των bits για κάθε σύμβολο, q η τάξη της

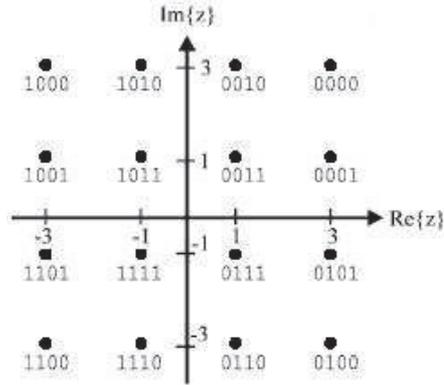
διαμόρφωσης. Άρα οι τιμές που μπορούν να πάρουν τα bits σε μια ακολουθία και τα σημεία ενός αστερισμού για QPSK, 16-QAM και 64-QAM είναι $2^2=4, 2^4=16$ και $2^6=64$ αντίστοιχα.

Κατά την απεικόνιση στο ορθοκανονικό σύστημα και με βάση την κωδικοποίηση Gray προκύπτουν τα παρακάτω διαγράμματα με τις ανάλογες τιμές.



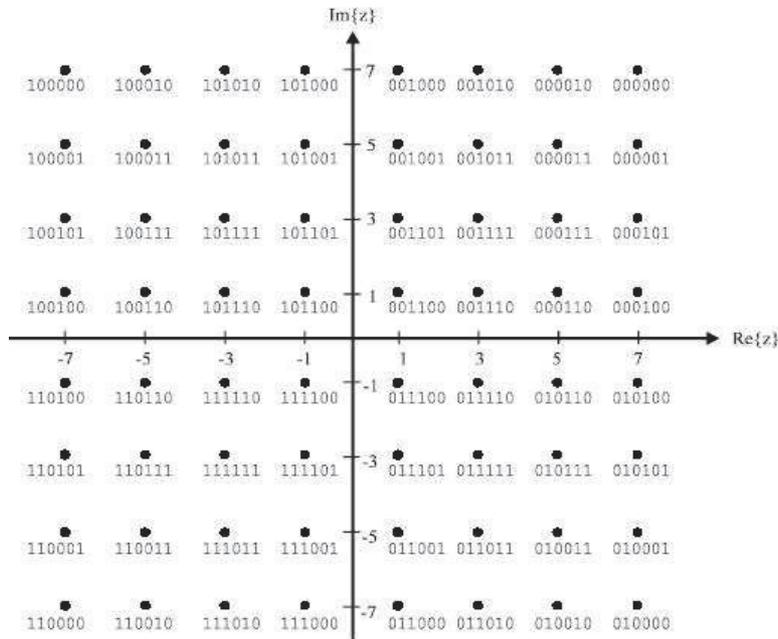
$$n \in \{-1, 1\}, m \in \{-1, 1\}$$

Σχ.2.7.1 Αστερισμός για QPSK (με $\alpha=1$)
QAM (με $\alpha=1$)



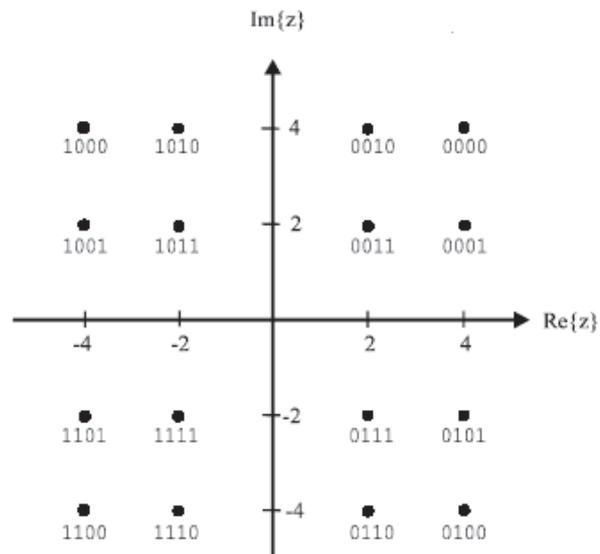
$$n \in \{-3, -1, 1, 3\}, m \in \{-3, -1, 1, 3\}$$

Σχ. 2.7.2 Αστερισμός για ομοιόμορφη 16-QAM



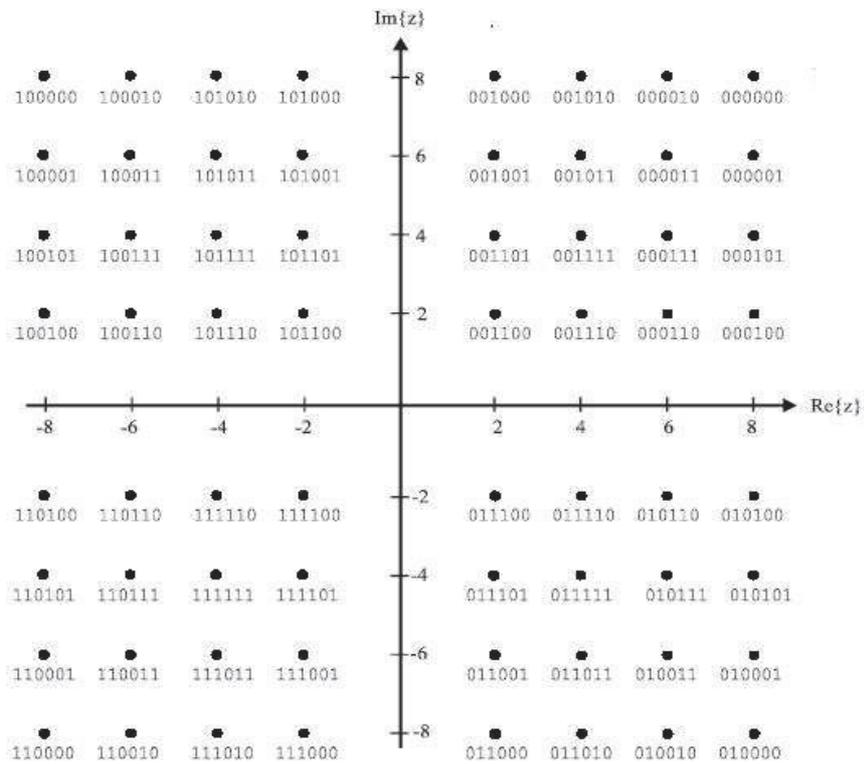
$$n \in \{-7, -5, -3, -1, 1, 3, 5, 7\}, m \in \{-7, -5, -3, -1, 1, 3, 5, 7\}$$

Σχ.2.7.3 Ομοιόμορφη 64-QAM (με $\alpha=1$)



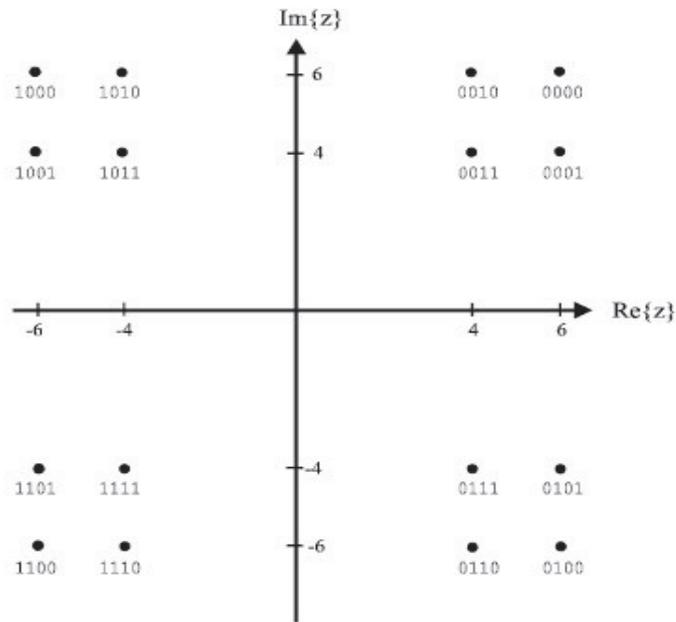
$$n \in \{-4, -2, 2, 4\}, m \in \{-4, -2, 2, 4\}$$

Σχ.2.7.4 Αστερισμός για Μη Ομοιόμορφη 16-QAM (με $\alpha=2$)



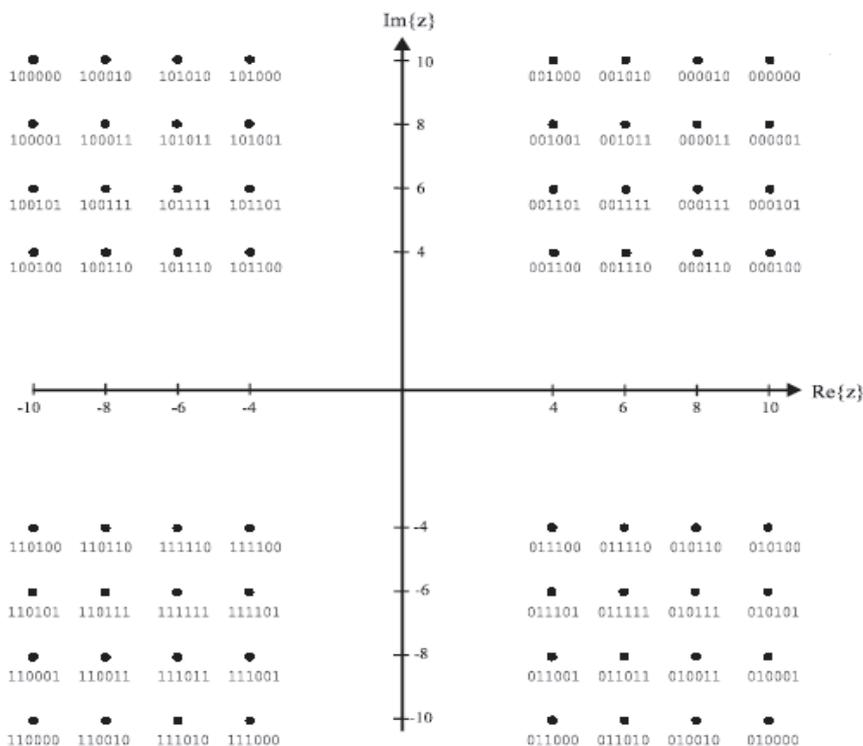
$$n \in \{-8, -6, -4, -2, 2, 4, 6, 8\}, m \in \{-8, -6, -4, -2, 2, 4, 6, 8\}$$

Σχ.2.7.5 Αστερισμός για Μη Ομοιόμορφη 64-QAM (με $\alpha=2$)



$$n \in \{-6, -4, 4, 6\}, m \in \{-6, -4, 4, 6\}$$

Σχ.2.7.6 Αστερισμός για Μη Ομοιόμορφη 16-QAM (με $\alpha=4$)



$$n \in \{-10, -8, -6, -4, 4, 6, 8, 10\}, m \in \{-10, -8, -6, -4, 4, 6, 8, 10\}$$

Σχ. 2.7.7 Αστερισμός για Μη Ομοιόμορφη 64-QAM (με $\alpha=4$)

Ο ρυθμός μετάδοσης μεταβάλλεται ανάλογα με το είδος διαμόρφωσης που θα επιλεγεί για τις

φέρουσες του OFDM, το ρυθμό εσωτερικής κωδικοποίησης και τη διάρκεια του διαστήματος ασφαλείας,. Χρησιμοποιούμε την QPSK διαμόρφωση σε περιπτώσεις όπου απαιτείται στιβαρή απόδοση σήματος, λόγω του χαμηλού ρυθμού δεδομένων που είναι διαθέσιμο ενώ, με την 64-QAM επιτυγχάνουμε υψηλότερους ρυθμούς δεδομένων.Μία τυπική επιλογή είναι η 64-QAM , με διάστημα ασφαλείας 1/32, και ρυθμό κωδικοποίησης 2/3 το οποίο μας δίνει ωφέλιμο φορτίο 24,13 Mbit/s.

QPSK: Οι ρυθμοί μετάδοσης που επιτυγχάνονται κυμαίνονται από 4,98 Mbps έως 10,56 Mbps

16-QAM: Οι ρυθμοί μετάδοσης που επιτυγχάνονται κυμαίνονται από 9,95 Mbps έως 21,11Mbps.

64-QAM: Οι ρυθμοί μετάδοσης που επιτυγχάνονται κυμαίνονται από 14,93 Mbps έως 31,67Mbps.

Modulation	Code rate	Guard interval			
		1/4	1/8	1/16	1/32
QPSK	1/2	4,98	5,53	5,85	6,03
	2/3	6,64	7,37	7,81	8,04
	3/4	7,46	8,29	8,78	9,05
	5/6	8,29	9,22	9,76	10,05
	7/8	8,71	9,68	10,25	10,56
16-QAM	1/2	9,95	11,06	11,71	12,06
	2/3	13,27	14,75	15,61	16,09
	3/4	14,93	16,59	17,56	18,10
	5/6	16,59	18,43	19,52	20,11
	7/8	17,42	19,35	20,49	21,11
64-QAM	1/2	14,93	16,59	17,56	18,10
	2/3	19,91	22,12	23,42	24,13
	3/4	22,39	24,88	26,35	27,14
	5/6	24,88	27,65	29,27	30,16
	7/8	26,13	29,03	30,74	31,67

Πίνακ.3 Διαθέσιμα bitrates (Mbit/s) για ένα σύστημα DVB-T σε κανάλι χωρητικότητας 8MHz

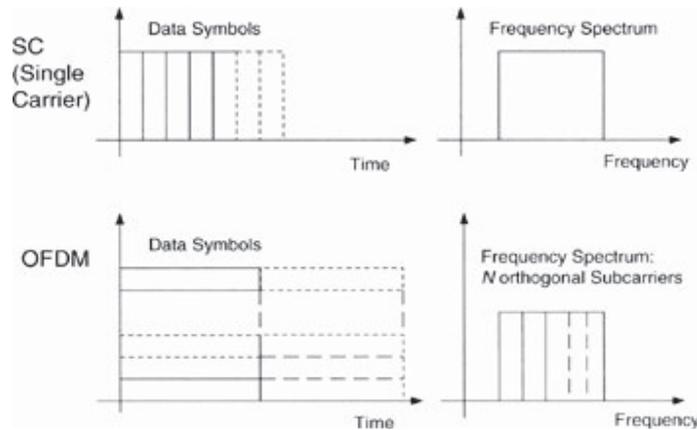
2.8.1 Αρχή Λειτουργίας OFDM

Κατά τη μετάδοση ενός σήματος σε ένα τηλεπικοινωνιακό κανάλι παρουσιάζεται το φαινόμενο της πολυόδευσης. Σύμφωνα με αυτό, όταν το σήμα συναντήσει εμπόδια όπως κτήρια, βουνα κτλ τότε αυτό ανακλάται με αποτέλεσμα ο δέκτης να λαμβάνει πολλαπλά αντίγραφα του ίδιου σήματος. Επίσης, ενισχύονται κάποιες συχνοτήτες και εξασθενούν κάποιες άλλες οδηγώντας σε υποβάθμιση της απόδοσης του δικτύου.

Για την επίλυση αυτού του προβλήματος χρησιμοποιείται η OFDM (Ορθογωνική Πολύπλεξη με Διαίρεση Συχνότητας). Το φάσμα διαιρείται σε πολλά υποκανάλια , τα δεδομένα θα μεταδοθούν ταυτόχρονα χρησιμοποιώντας ένα πλήθος από υποφέροντα (sub-carriers) τα οποία είναι ομοίμορφα κατανεμημένα στο πεδίο των συχνοτήτων. Τα δεδομένα σε κάθε υποφέρον διαμορφώνονται με QPSK, 16-QAM, 64-QAM αναλόγως με το BER (Bit Error Rate) και το ρυθμό μετάδοσης που θέλουμε να πετύχουμε.. Οι συχνοτήτες τους επιλέγονται κατάλληλα ώστε να είναι ορθογώνιες μεταξύ τους χωρίς να παρεμβάλει η μία την άλλη. Επίσης, αφού κάθε υποφέρον μεταφέρει μόνο ένα μέρος της πληροφορίας, η διάρκεια των συμβόλων αυξάνεται κάτι που

καθιστά το σήμα λιγότερο ευαίσθητο στην διασυμβολική παρεμβολή.

Αν υπάρξουν διαλείψεις αυτές θα επηρεάσουν κάποια υποφέροντα και όχι ολόκληρο το σήμα. Άρα, η πληροφορία που θα χαθεί, θα είναι αυτή που μεταφέρεται απο τα συγκεκριμένα υποφέροντα, και μπορεί να ανακτηθεί απο τους κωδικες διόρθωσης σφαλμάτων.

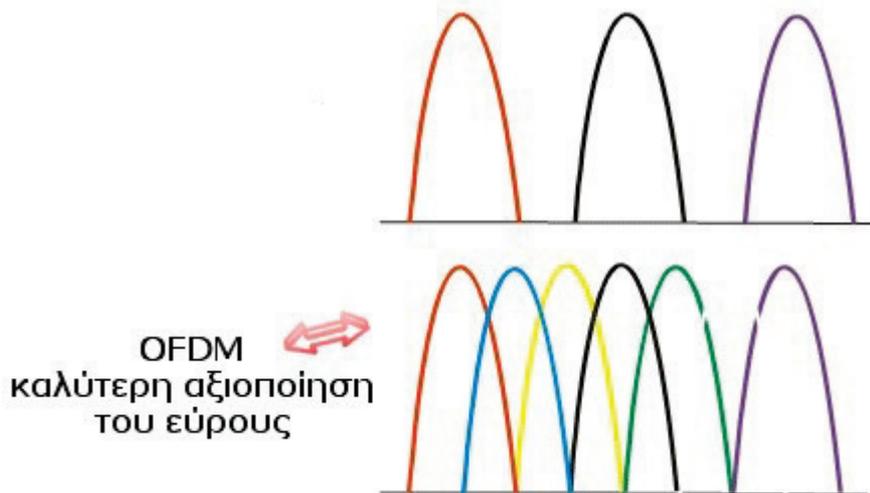


Σχ. 2.8.1 Αναπαράσταση της συχνότητας σε συνάρτηση με το χρόνο μόνου φέροντος και OFDM

Η τεχνική της OFDM εννοεί τον σχεδιασμό δικτύων μονής συχνότητας (SFNs). Σε αυτά τα δίκτυα ο δέκτης λαμβάνει το σήμα από το σύνολο των σημάτων που δέχεται από γειτονικούς πομπούς. Έτσι πετυχαίνουμε μεγαλύτερη κάλυψη και εξοικονόμηση συχνοτήτων. Η εκπομπή ενός OFDM συστήματος υλοποιείται με αντίστροφο μετασχηματισμό Fourier (IFFT - Inverse Fast Fourier Transform) ενώ στη λήψη με FFT (Fast Fourier Transform).

Ορίζονται δύο τρόποι λειτουργίας: η λειτουργία 2K (χρησιμοποιώντας 1704 φορείς) και η 8K (χρησιμοποιώντας 6816 φορείς). Η κύρια διαφορά των δυο λειτουργιών είναι στην απόδοση. Η λειτουργία 8K υποστηρίζει διαστήματα ασφαλείας τέσσερις φορές μεγαλύτερα από αυτά της 2K. Οπότε μπορεί να ανεχτεί στη λήψη ισχυρή ηχώ η οποία είναι τέσσερις φορές μεγαλύτερη από τη μέγιστη ηχώ που μπορεί να αντεπεξέλθει η 2K λειτουργία.

Η λειτουργία «2K» είναι κατάλληλη για λειτουργία ενός πομπού και για μικρά δίκτυα μιας συχνότητας τα οποία έχουν περιορισμένο εύρος κάλυψης. Η λειτουργία «8K» μπορεί να χρησιμοποιηθεί τόσο για λειτουργία ενός πομπού όσο και για μικρά και μεγάλα δίκτυα μονής συχνότητας.



Σχ.2.8.2 Εξοικονόμηση εύρους φάσματος με την τεχνική OFDM

2.8.2 Δομή ενός OFDM πλαισίου

Το εκπεμπόμενο σήμα στο DVB-T είναι οργανωμένο σε πλαίσια κάθε ένα από αυτά αποτελείται από 68 COFDM (Coded Orthogonal Frequency Division Multiplexing) σύμβολα με διάρκεια T_F . Ένα υπερπλαίσιο αποτελείται από 4 πλαίσια. Κάθε σύμβολο αποτελείται από ένα σύνολο $K=6817$ και $K=1705$ φερόντων στη λειτουργία 8K και 2K αντίστοιχα, τα οποία μεταδίδονται με διάρκεια T_s . Το πλαίσιο αποτελείται από δύο μέρη: το πρώτο είναι το χρήσιμο τμήμα με διάρκεια T_U και το δεύτερο είναι ένα διάστημα ασφαλείας με διάρκεια Δ . Αυτό το διάστημα ασφαλείας αποτελεί μια κυκλική συνέχιση του χρήσιμου τμήματος T_U και θα προστεθεί πριν από αυτό. Αναλυτικές πληροφορίες αναφέρονται στο Παράρτημα Α

Τα σύμβολα σε ένα πλαίσιο OFDM είναι αριθμημένα από το 0 έως το 67 και περιέχουν δεδομένα καθώς και πληροφορίες αναφοράς. Λόγω ότι το σήμα OFDM περιέχει πολλά διαμορφωμένα φέροντα, κάθε σύμβολο είναι δυνατόν να διαιρεθεί σε κελιά. Καθένα από τα οποία αντιστοιχίζεται στη διαμόρφωση που επήλθε σε ένα φέρον κατά τη διάρκεια του συμβόλου.

2.8.3 Σήματα με πληροφορίες αναφοράς

Σε κάθε OFDM σύμβολο περιέχονται εκτός από τα δεδομένα και πληροφορίες αναφοράς που λέγονται σύμβολα πιλότοι. Αυτά τα σύμβολα πιλότοι αποτελούν την ταυτότητα του διαμορφωμένου σήματος και χρησιμοποιούνται από το δέκτη ούτως ώστε να μπορέσει να τα

επεξεργαστεί.. Τα σύμβολα διακρίνονται σε:

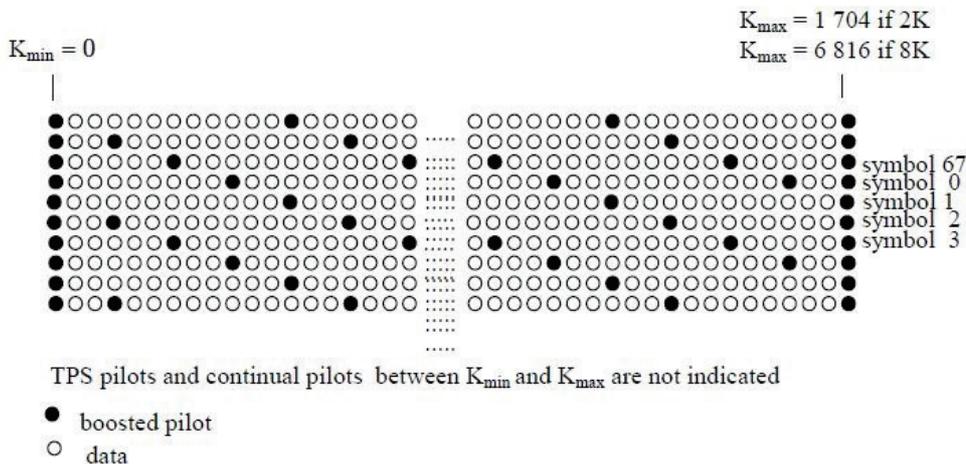
1. **Διάσπαρτα φέροντα πιλότοι:** Αυτά εκπέμπουν εκ περιτροπής ένα σήμα γνωστό στο δέκτη. Μετρώντας την ισχύ αυτού του σήματος ο δέκτης οποιαδήποτε στιγμή μπορεί να κάνει εκτίμηση της απόκρισης συχνότητας του διαύλου.
2. **Συνεχόμενα φέροντα πιλότοι:** Έχουν σταθερή θέση μέσα στο σύμβολο και χρησιμοποιούνται κατά τη διάρκεια του συγχρονισμού.
3. **Φέροντα σηματοδοσίας παραμέτρων μετάδοσης TPS:** Αυτά αποστέλλουν τις παραμέτρους του εκπεμπόμενου σήματος στο δέκτη που αφορούν τη κωδικοποίηση του καναλιού και τη διαμόρφωση.

Οι πληροφορίες με τις παραμέτρους αυτές είναι:

- Η διαμόρφωση και η τιμή του α .
- Η επιλογή ιεραρχικού ή μη ιεραρχικού τρόπου μετάδοσης.
- Το διάστημα διαφύλαξης (guard interval).
- Οι εσωτερικοί ρυθμοί μετάδοσης.
- Ο τρόπος μετάδοσης (2K ή 8K).
- Ο αριθμός του πλαισίου στο υπέρ-πλαίσιο.
- Η ταυτοποίηση κελιού.

Αναλυτικές πληροφορίες για τη δομή ενός TPS block περιγράφονται στο Παράρτημα Β.

Ο δέκτης με τη σειρά του συγχρονίζει, ισοσταθμίζει, αποκωδικοποιεί το σήμα ώστε να αποκτήσει πρόσβαση στις πληροφορίες που περιέχονται στους πιλότους TPS.



Σχ. 2.9.1 Δομή του εκπεμπόμενου πλαισίου, μετά την εισαγωγή των συμβόλων-πιλότων

Οι πληροφορίες αυτές θα σταλούν και στο επόμενο πλαίσιο ούτως ώστε να ενημερωθεί ο δέκτης για τυχόν αλλαγές (θεωρείται δεδομένο ότι οι αρχικές τιμές είναι γνωστές).

Continual pilot carrier positions (index number k)													
2K mode							8K mode						
0	48	54	87	141	156	192	0	48	54	87	141	156	192
201	255	279	282	333	432	450	201	255	279	282	333	432	450
483	525	531	618	636	714	759	483	525	531	618	636	714	759
765	780	804	873	888	918	939	765	780	804	873	888	918	939
942	969	984	1 050	1 101	1 107	1 110	942	969	984	1 050	1 101	1 107	1 110
1 137	1 140	1 146	1 206	1 269	1 323	1 377	1 137	1 140	1 146	1 206	1 269	1 323	1 377
1 491	1 683	1 704					1 491	1 683	1 704	1 752	1 758	1 791	1 845
							1 860	1 896	1 905	1 959	1 983	1 986	2 037
							2 136	2 154	2 187	2 229	2 235	2 322	2 340
							2 418	2 463	2 469	2 484	2 508	2 577	2 592
							2 622	2 643	2 646	2 673	2 688	2 754	2 805
							2 811	2 814	2 841	2 844	2 850	2 910	2 973
							3 027	3 081	3 195	3 387	3 408	3 456	3 462
							3 495	3 549	3 564	3 600	3 609	3 663	3 687
							3 690	3 741	3 840	3 858	3 891	3 933	3 939
							4 026	4 044	4 122	4 167	4 173	4 188	4 212
							4 281	4 296	4 326	4 347	4 350	4 377	4 392
							4 458	4 509	4 515	4 518	4 545	4 548	4 554
							4 614	4 677	4 731	4 785	4 899	5 091	5 112
							5 160	5 166	5 199	5 253	5 268	5 304	5 313
							5 367	5 391	5 394	5 445	5 544	5 562	5 595
							5 637	5 643	5 730	5 748	5 826	5 871	5 877
							5 892	5 916	5 985	6 000	6 030	6 051	6 054
							6 081	6 096	6 162	6 213	6 219	6 222	6 249
							6 252	6 258	6 318	6 381	6 435	6 489	6 603
							6 795	6 816					

Σχ. 2.9.2 Υπόκανάλια που περιέχουν συνεχή σύμβολα-πιλότους

2.9 Μετατροπείας Ψηφιακού σε Αναλογικό (DAC)

Είναι το τελικό στάδιο, στο οποίο το ψηφιακό σήμα μετατρέπεται σε αναλογικό σήμα στη συνέχεια διαμορφώνεται στις συχνότητες των (VHF, UHF) . Το εύρος ζώνης είναι σχεδιασμένο να προσαρμόσει κάθε σήμα DVB-T σε κανάλια των 6,7,8, MHz.

ΚΕΦΑΛΑΙΟ 3^ο ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΜΕΘΟΔΟΥ ΣΕ ΓΛΩΣΣΑ C

3.1 Αρχείο “COFDM-Rayleigh.cpp”:

Στη γραμμή 18 ορίζεται ένας timer.

Στη γραμμή 18 δηλώνεται μια σταθερά με όνομα `m` με τιμή 4096.

Στη γραμμή 19 δηλώνεται μια σταθερά με όνομα `lsnr` με τιμή 25.

Στη γραμμή 20 δηλώνεται μια σταθερά με όνομα `q` με τιμή 64.

Στη γραμμή 21 δηλώνεται η σταθερά `pi`.

Στη γραμμή 22 δηλώνεται η σταθερά `eps` με τιμή 0.00000001. (Χρησιμοποιείται για μεγαλύτερη ακριβεία στη σύγκριση αποτελεσμάτων μεταξύ του matlab και της C++).

Στη γραμμή 25 δηλώνεται το πλήθος των επαναλήψεων.

Στη γραμμή 26 δηλώνεται ένας ακέραιος πίνακας μεγέθους `lsnr`.

Στη γραμμή 27 δηλώνεται ένας πίνακας τύπου `double`, μεγέθους `lsnr`, του οποίου τα στοιχεία θα απεικονίζονται στη γραφική παράσταση.

Στη γραμμή 30 δηλώνεται μια ακέραια μετβλητή `k` που μετατρέπεται σε `unsigned integer` και υπολογίζεται η τετραγωνική ρίζα του πίνακα `q` προκειμένου να χρησιμοποιηθεί στις συναρτήσεις `void qammod` και `void qamdemod`.

Στη γραμμή 35 ορίζεται μια δομή `struct cmplx { double re , im ; } ;` που θα χρησιμοποιηθεί στους μιγαδικούς αριθμούς.

Στη γραμμή 39 δημιουργούμε μια συνάρτηση που δημιουργεί τον πίνακα `data_in` και χρησιμοποιείται για την εισαγωγή των δεδομένων στο συστήμα μας.

Στι γραμμές 42 έως 43 δηλώνονται τα `generator polynomilas`.

Στη γραμμή 46 δηλώνεται η σταθερά `const int mm` που ορίζει το πλήθος των καταχωρητών που χρησιμοποιεί ο `convolutional encoder`.

Στη γραμμή 49 δηλώνεται ένας πίνακας τύπου `int`, ο `state[mm]` και αρχικοποιείται.

Στη γραμμή 47 δηλώνεται το μήκος της της εισόδου (χρησιμοποιείται ρυθμός κωδικοποίησης $\frac{1}{2}$).

Στη γραμμή 53 εισάγονται στον πίνακα `input[x]` ψευδοτυχαία, τα στοιχεία με εύρος τιμών `[0,1]`.

Στη γραμμή 66 πραγματοποιείται πολλαπλασιασμός μεταξύ των στοιχείων του πίνακα `gen` και του `tempx`.

Στη γραμμή 73 εκτελείται η πράξη XOR μεταξύ του πίνακα `tempy` με το `z`.

Στις γραμμές 78 τα στοιχεία του πίνακα `state` μετακινούνται μια θέση προς τα δεξιά.

Στη γραμμή 79 τοποθετείται στην πρώτη θέση (που έχει μείνει κενή) το καινούριο στοιχείο.

Στις γραμμές 82 έως 83 για κάθε επανάληψη της `for` προσθέτονται δυο καινούρια στοιχεία στο `data_in` τα οποία προέρχονται από ένα στοιχείο του πίνακα `input`.

Στις γραμμές 89 έως 148 υλοποιείται η συνάρτηση `void qammod(int x[m], cmplx y[m])` για την QAM διαμόρφωση. Δημιουργούνται δυο ακέραιοι πίνακες, στον πίνακα `x` θα αποθηκευτούν οι πραγματικές τιμές του σήματος και στον `y` οι φανταστικές. Στη συνέχεια, οι δυο αυτοί πίνακες προστίθενται για να δημιουργήσουν έναν νέο πίνακα `c` ο οποίος θα περιέχει τα στοιχεία που αποτελούν τα σημεία αστερισμού της διαμόρφωσης.

Στις γραμμές 175 έως 219 υλοποιείται η συνάρτηση `void fft(double data[], int nn, int isign)` όπως ορίζεται στο βιβλίο *Numerical Recipes in C* στις σελίδες 507-508.

Στις γραμμές 223 έως 232 υλοποιείται η συνάρτηση `cmplx mean_cv(cmplx y[], int n)` για τον υπολογισμό της μέσης τιμής.

Στη γραμμή 236 έως 246 υλοποιείται η συνάρτηση `double std_var(cmplx y[], int n)` που υπολογίζει την τυπική απόκλιση του σήματος.

Στη γραμμή 249 υλοποιείται η συνάρτηση `void add_white_noise(cmplx y[], int snr)` που χρησιμοποιείται για να την προσθήκη του Additive white Gaussian noise (AWGN).

Στη γραμμή 255 υπολογίζεται το επίπεδο του θορύβου.

Στις γραμμές 261 έως 306 περιγράφεται και υλοποιείται μια μέθοδος σύμφωνα με την οποία χρησιμοποιούμε ομοιόμορφα κατανομημένες μεταβλητές ώστε να παράγουμε κανονικές τυχαίες μεταβλητές (http://en.wikipedia.org/wiki/Box%E2%80%93Muller_transform).

Στη γραμμή 290 γίνεται ο πολλαπλασιασμός της σταθεράς `RAND_MAX` με το 1.0, το αποτέλεσμα μετατρέπεται από τη C++ σε πραγματικό ώστε να γίνει κανονική διαίρεση και όχι ακέραια. Στη συνέχεια, αφού η `u1` παίρνει ομοιόμορφα, ακέραιες τιμές στο διάστημα `[0,RAND_MAX]`, διαιρώντας με `RAND_MAX` μετατρέπεται σε πραγματική ομοιόμορφη στο διάστημα `[0,1]`, που είναι αυτό που χρειάζεται για τις εξισώσεις Box-Muller.

Στις γραμμές 293 έως 296 δημιουργείται το μοντέλο του (Rayleigh) θορύβου.

Στη γραμμή 298 και 300 προστίθεται ο θόρυβος στο μεταδιδόμενο σήμα.

Στις γραμμές 309 έως 363 υλοποιείται η συνάρτηση `void qamdmod(cmplx y[m], int x[m])` για την QAM αποδιαμόρφωση.

Στις γραμμές 319 έως 325 δημιουργείται το πλήθος των μιγαδικών αριθμών.

Στις γραμμές 349 έως 353 γίνεται η εύρεση του ελάχιστου μιγαδικού.

Στις γραμμές 355 έως 361 εντοπίζουμε την ακριβή θέση στην οποία βρίσκεται ο μιγαδικός αριθμός.

Στη γραμμή 373 αρχικοποιείται η τιμή της ψευδοτυχαίας ακολουθίας.

Στη γραμμή 378 πραγματοποιείται η αρχικοποίηση των πινάκων snr και rate0.

Στη γραμμή 383 καλείται η συνάρτηση `construct_data_in`.

Στη γραμμή 397 καλείται η συνάρτηση `qammod (data_in , data_mod)`.

Στις γραμμές 401 έως 402 πραγματοποιείται η αποθήκευση των μιγαδικών αριθμών με την ακόλουθη μορφή, `data[2*n+1] = real(x(n))` για το πραγματικό μέρος και `data[2*n+2] = imag(x(n))` για το φανταστικό μέρος.

Στη γραμμή 405 καλείται η συνάρτηση `fft(data , m , -1)` για τον υπολογισμό του Inverse Fast Fourier Transform. Όπου `data`, ένας πίνακας τύπου `double` του οποίου το μέγεθος είναι ίσο με `NFFT*2+1`.

Στις γραμμές 413 έως 418 εισάγεται το Guard Interval στο ωφέλιμο σήμα.

Στη γραμμή 421 καλείται η συνάρτηση `add_white_noise (data_guard , ebn0)`.

Στις γραμμές 428 έως 430 αφαιρείται το Guard Interval απο το ωφέλιμο σήμα.

Στη γραμμή 493 καλείται η συνάρτηση `fft (data , m , 1)` για τον υπολογισμό του Fast Fourier Transform. Όπου `data`, ένας πίνακας τύπου `double` του οποίου το μέγεθος είναι ίσο με `NFFT*2+1`.

Στις γραμμές 442 έως 444 διαβάσουμε τις τιμές που προκύπτουν από τον υπολογισμό του Fast Fourier Transform απο το τέλος προς την αρχή. Λόγω του οτι η κανονικοποίηση του δεν πραγματοποιείται μέσα στη συνάρτηση `fft` διαιρούμε το πραγματικό και το φανταστικό μέρος των μιγαδικών αριθμών που με το `m`.

Στη γραμμή 446 καλείται η συνάρτηση `qamdemod(data_rem_all_guard , data_out)`.

Στη γραμμή 452 γίνεται σύγκριση μεταξύ των πινάκων `data_in` και `data_out` και υπολογίζονται οι διαφορές.

Στη γραμμή 456 πολλαπλασιάζουμε τις `diff_num` και `m` με το 1.0 για να μετρατρέψουμε το αποτέλεσμα της διαίρεσης σε πραγματικό. Όπου `diff_num` ορίζεται μια ακέραια μεταβλητή για τις τιμές που περιέχουν τις λάθος τιμές. Οπότε ο λόγος του `diff_num` με το `m` υπολογίζει το ρυθμό σφαλμάτων (BER).

Στις γραμμές 466 έως 468 υπολογίζονται τα στοιχεία του πίνακα `rate0` που θα χρησιμοποιηθούν για την απεικόνιση της γραφικής παράστασης.

Στις γραμμές 469 έως 471 υπολογίζεται ο χρόνος εκτέλεσης του προγράμματος.

Στις γραμμές 486 έως 502 δημιουργείται η γραφική παράσταση με τη χρήση του `gnuplot`.

3.2 Πηγαίος κώδικας MATLAB –Αρχείο “OFDM_Noise_k.m”:

```
% Initialization
M = 8096;          % Number of subchannels
Guard_Interval = M/4;% Guard interval length
q = 64;           % Modulation level
nloop = 100;      % Number of iteration
SNR= 0:25;        % signal to noise ratio vector in dB
Rate0= zeros(1, length(SNR)); % initializing bit error rate
%%
for( ebn0 = 1 : length(SNR))
    ebn0
    % for(i = 1 : nloop)
%% Transmitter
        Data_In = randint(M,1,q);    %% Data generation
        %% Modualtion
        Data_Mod = my_qammod(Data_In,q);
        %% IFFT
        Data_IFFT = ifft(Data_Mod,M);
        %% Guard interval insertion
        Data_Guard = [Data_IFFT(M- Guard_Interval + 1 : M);Data_IFFT];
        %% P/S conversion
        [a b]=size(Data_Guard);
        Data_Tx=reshape(Data_Guard,a*b,1);
%% Channel Effect%% AWGN
        Data_Noise = awgn(Data_Tx,ebn0,'measured');

%% Receiver
        %% S/P conversion
        Data_Rx=reshape(Data_Noise,a,b);
        %% Guard interval removal
        Data_Removeal_Gaurd = Data_Rx(Guard_Interval+1:M+Guard_Interval);
        %% FFT
        Data_FFT = fft(Data_Removeal_Gaurd,M);
        %% Demodualtion
        Data_Out = my_qamdemod(Data_FFT,q);
%% BER
        %
        [nErr rate0] = symerr(Data_Out,Data_In);
        Rate0(ebn0)= Rate0(ebn0) + rate0;
    end
    Rate0(ebn0)= Rate0(ebn0)/nloop;
%end

hold on
f1 = figure(1);
set(f1,'color',[1 1 1]);
hold on
```

```
semilogy(SNR,Rate0,'r-o')  
xlabel('Signal to Noise Ratio [SNR] in dB');  
ylabel('Bit Error Rate [BER]' %legend('OFDM'))
```

3.3 Πηγαίος κώδικας MATLAB –Αρχείο “ my_qammod.m”:

```
function y = my_qammod(x, m)
%#codegen
%here y is the QAM modulated output in vector form like x+jy, x is the
%input vector which is in the range of 0 to M-1, where M is the type of M-QAM
%which is proportional to the power of 2 (2 =2^1, 4 =2^2, 8 = 2^3..)

%santosh shah, The LNM IIT Jaipur (India)(santosh.jnt@gmail.com) 23/04/07

%i have to check if M is equal to the power of 2 and x is in between 0 to
%M-1 so..
if log2(sqrt(m))~= floor(log2(sqrt(m)))
    error('Please check the value of m that you have provided for type M-QAM.');
```

```
end

%i need to remove (x == floor(x)), if we want the value of x is in between anywhere 0 to M-1
if ~all(x >= 0 & x < m & x == floor(x))
    error('Please check the messege vector it must be in between 0 to M-1.');
```

```
end

%now generating the cancellation points

k = sqrt(m);
r = 2*(0:k-1) - k + 1;
[xi, yi] = meshgrid(r);
c = xi + 1j*flipud(yi);

% now calling the corresponding element from vector c as x.
y = c(x+1);
```

3.4 Πηγαίος κώδικας MATLAB –Αρχείο “ my_qamdemod.m”:

```
function z = my_qamdemod(y, m)
%#codegen

%this is the function for demodulation of QAM, y is the M-QAM Modulated
%signam, m is playing the same role. z is the demodulated output. input y
%is a row or coulumn vector and, z should be the row vector.

%santosh shah, The LNM IIT Jaipur (India)(santosh.jnt@gmail.com) 23/04/07

% again i need to check the value of M, so by the same process.
if log2(sqrt(m))~= floor(log2(sqrt(m)))
    error('Please check the value of m that you have provided for type M-QAM.');
```

```
end

%taking the reverse process
k = sqrt(m);
r = 2*(0:k-1) - k + 1;
[xi, yi] = meshgrid(r);
c = xi + 1j*flipud(yi);
c = c(:);

%now comparing the data from c's vector after rounding the input data.
% Allocate space for output
z = zeros(size(y));

% Slicer: Find closest constellation symbol, symbol-by-symbol.
for k = 1:length(y)
    [nil ind] = min(abs(y(k) - c));
    z(k) = ind - 1;
end
```

3.5 Πηγαίος κώδικας σε C++

```
1 // *****
2 // Onoma : GKOUVAS KONSTANTINOS KONSTANTINOS
3 // File : COFDM-Rayleigh.cpp
4 // Descr : OFDM test
5 // Date : 04-12-2013
6 // *****
7
8 #include <iostream>
9 #include <cstdlib> /*rand,srand */
10 #include <cstdio> /* fprintf,popen */
11 #include <cmath> /*sqrt, M_PI*/
12 #include <ctime> /*normal_distribution */
13
14 using namespace std ;
15
16 clock_t startTime = clock(); // orismos enos timer
17
18 #define m 4096 /* mege8os simatos
19 #define lsnr 25 /*epipedo tou SNR
20 #define q 64 /* taxi tis QAM
21 #define pi M_PI
22 #define eps 0.00000001
23
24 int gi=1/32, guard_interval = m*gi , dim=m+m*gi;
25 int nloop = 100 ;
26 int snr[lsnr] ;
27 double rate0[lsnr] ;
28
29
30 int k = static_cast<unsigned int>(sqrt(q) + 0.001) ; // xrisimopoihsi stis
31 sinartiseis gia diamorfosi kai apodiamorfosi
32
33 int data_in[m] , data_out[m] ;
34
35 struct cmplx { double re , im ; } ;
36
37 /** dimioyrgia tou pinaka data_in gia eisagogi dedomenon sto sistima ***/
38
39 void construct_data_in(void) {
40
41 /**dilosti ton generator polynomials***/
42 const int nn=2 , KK=7 ;
43 const int gen[nn][KK] = {{1,1,1,1,0,0,1},{1,0,1,1,0,1,1}} ;
44
45 // arithmos kataxoriton
46 const int mm=KK-1 ;
47
48 // arxiki katastasi ton kataxoriton
49 int state[mm]={0,0,0,0,0,0} ;
50
51 //orizetai to mikos tou encoder
```

```

52
53 const int hh=m/2 ;
54 int inputx[hh] ;
55
56 // gemisma tou inputx me tixaia bits
57
58 for(int x=0 ; x<hh ; x++) inputx[x]=rand()%2;
59
60 for(int x=0 ; x<hh ; x++) {
61     int tempx=inputx[x] , tempy[nn] ;
62
63     for(int i=0 ; i<nn ; i++) {
64
65         // pollaplasiasmos ton stoixeion tou gen me ayta tou tempx
66         tempy[i]=gen[i][0]*tempx ;
67
68         for(int j=1 ; j<KK ; j++) {
69
70             // XOR toy pinaka tempy me to z
71
72             int z=gen[i][j]*state[j-1] ;
73             tempy[i]=tempy[i]^z ; } }
74
75     // ta stoixeia metakinonuntai mia thesi pros ta dexia
76     // topothetisi tou neou stoixeiou stin keni thesi
77     for(int i=mm-1 ; i>0 ; i--)
78         state[i]=state[i-1] ;
79     state[0]=tempx ;
80 //topothesisi 2 neon stoixeion sto data_in proerxomena apo ton pinaka input
81
82     data_in[2*x] = tempy[0] ;
83     data_in[2*x+1]= tempy[1] ; }
84 return ; }
85
86 /*** complex modulus for use in qamdemod ***/
87 double modul(cmplx z) { return sqrt(z.re*z.re+z.im*z.im) ; }
88
89 void qammod(int x[m] , cmplx y[m]) {
90 int r[k] , i , j , l , xi[k][k] , yi[k][k] ;
91 cmplx c[q] ;
92
93     /* obolete -> is in definitions */
94     /* k = sqrt(q)
95     /*k = sqrt(q) ; */
96
97     /* r=2*(0:k-1) - k + 1 */
98     for(i=0 ; i<k ; i++) r[i]=2*i-k+1 ;
99
100     /*****
101     /***** test r=2*(0:k-1) - k + 1 *****/
102     /*
103         for(i=0 ; i<k ; i++) cout << r[i] << " " ; cout << endl << endl ;
104     */
105     /*****
106
107     /* [xi,yi] = meshgrid(r) */
108     for(i=0 ; i<k ; i++)
109         for(j=0 ; j<k ; j++) {
110             xi[i][j] = r[j] ;

```

```

111         yi[i][j] = r[i] ; }
112
113  /*****
114  /***** test [xi,yi] = meshgrid(r) *****/
115  /*
116      for(i=0 ; i<8 ; i++) {
117          for(j=0 ; j<8 ; j++) {
118              cout << xi[i][j] << " " ; }
119          cout << endl ; }
120      cout << endl ;
121
122      for(i=0 ; i<8 ; i++) {
123          for(j=0 ; j<8 ; j++) {
124              cout << yi[i][j] << " " ; }
125          cout << endl ; }
126      cout << endl ;
127  */
128  /*****/
129
130      /* c=xi+l*j*flipud(yi) */
131      for(l=0 ; l<q ; l++) {
132          i = l/k ; j=l%k ;
133          c[l].re = r[i] ;
134          c[l].im = r[k-j-1] ; }
135
136  /*****
137  /***** test c=xi+l*j*flipud(yi) *****/
138  /*
139      for(l=0 ; l<q ; l++) {
140          cout << c[l].re << "+i " << c[l].im << endl ; }
141      */
142  /*****/
143
144      for(l=0 ; l<m ; l++) {
145          y[l].re = c[x[l]].re ;
146          y[l].im = c[x[l]].im ; }
147
148      return ; }
149
150
151  /* ilopoihsi tis FFT/IFFT routine , opos orizetai sto Numerical Recipes in C)
152
153  Inputs:
154      data[] : array of complex* data points of size 2*NFFT+1.
155      data[0] is unused,
156      * the n'th complex number x(n), for 0 <= n <= length(x)-1, is stored
157  as:
158          data[2*n+1] = real(x(n))
159          data[2*n+2] = imag(x(n))
160      if length(Nx) < NFFT, the remainder of the array must be padded with
161  zeros
162
163      nn : FFT order NFFT. This MUST be a power of 2 and >= length(x).
164      isign: if set to 1,
165              computes the forward FFT
166              if set to -1,
167              computes Inverse FFT - in this case the output values
168  have
169              to be manually normalized by multiplying with 1/NFFT.

```

```

170  Outputs:
171      data[] : The FFT or IFFT results are stored in data, overwriting the
172  input.
173  */
174
175  void fft(double data[], int nn, int isign) {
176
177      int n, mmax, s, j, istep, i;
178      double wtemp, wr, wpr, wpi, wi, theta;
179      double tempr, tempi;
180
181      n = nn << 1;
182      j = 1;
183      for (i = 1; i < n; i += 2) {
184          if (j > i) {
185              tempr = data[j];      data[j] = data[i];      data[i] = tempr;
186              tempr = data[j+1]; data[j+1] = data[i+1]; data[i+1] = tempr;
187          }
188          s = n >> 1;
189          while (s >= 2 && j > s) {
190              j -= s;
191              s >>= 1;
192          }
193          j += s;
194      }
195      mmax = 2;
196      while (n > mmax) {
197          istep = 2*mmax;
198          theta = 2*pi/(isign*mmax);
199          wtemp = sin(0.5*theta);
200          wpr = -2.0*wtemp*wtemp;
201          wpi = sin(theta);
202          wr = 1.0;
203          wi = 0.0;
204          for (s = 1; s < mmax; s += 2) {
205              for (i = s; i <= n; i += istep) {
206                  j = i + mmax;
207                  tempr = wr*data[j] - wi*data[j+1];
208                  tempi = wr*data[j+1] + wi*data[j];
209                  data[j] = data[i] - tempr;
210                  data[j+1] = data[i+1] - tempi;
211                  data[i] += tempr;
212                  data[i+1] += tempi;
213              }
214              wr = (wtemp = wr)*wpr - wi*wpi + wr;
215              wi = wi*wpr + wtemp*wpi + wi;
216          }
217          mmax = istep;
218      }
219  return ; }
220
221  /**/ ypologismos mesis timis tou simatos ***/
222
223  cmplx mean_cv(cmplx y[], int n) {
224  cmplx z ;
225  z.re = z.im = 0 ;
226
227  for(int ii=0 ; ii<n ; ii++) {
228      z.re += y[ii].re ;

```

```

229     z.im += y[ii].im ; }
230
231 z.re /= n ; z.im /= n ;
232 return z ; }
233
234
235 /** ypologismos tipikis apoklisis simatos */
236 double std_var(cmplx y[] , int n) {
237 cmplx mn = mean_cv(y,n) , z ;
238 double res=0 ;
239
240 for(int ii=0 ; ii<n ; ii++) {
241     z.re = y[ii].re - mn.re ;
242     z.im = y[ii].im - mn.im ;
243     res += z.re * z.re + z.im * z.im ; }
244 res/=(n-1) ;
245
246 return sqrt(res) ; }
247
248 /** prosthiki AWGN thorivou sto sima */
249 void add_white_noise(cmplx y[] , int snr) {
250
251     /** ypologismos tipikis apoklisis */
252     double stds = std_var(y , dim) ;
253     double sig_in_db = 20*log10(stds) ;
254     double noise_in_db = sig_in_db - snr ;
255     double scalar = exp(noise_in_db*log(10)/20) ; //ypologizetai to epipedo toy
256 8orivou
257
258
259
260
261     double u1_c , u2_c , r_c , th_c , z1_c ;
262     u1_c = static_cast<double>(rand()) ; u2_c = static_cast<double>(rand()) ;
263     z1_c = r_c*cos(th_c) ;
264
265 for(int ii=0 ; ii<dim ; ii++) {
266
267     /** c/c++ supports pseudo-random uniform variables */
268     /** xrisimopoietai i exisosi Box-Muller gia tin kataskeui kanonikis
269 anexartitis
270 metavlitis apo dio omoiomorfes metavlites sto diastima 0,1). */
271     /** http://en.wikipedia.org/wiki/Box%E2%80%93Muller_transform */
272
273
274
275     /** dimiourgia tou Rayleigh fading */
276     double u1_t , u2_t , r_t , th_t , z1_t ;
277     u1_t = static_cast<double>(rand()) ; u2_t =
278 static_cast<double>(rand()) ;
279     z1_t = sqrt(1/2)*r_t * cos(th_t) ;
280
281     double real , imag ;
282     real = z1_t*y[ii].re-z1_c*y[ii].im ;
283     imag = z1_c*y[ii].re+z1_t*y[ii].im ;
284
285
286     double u1 , u2 , r , th , z1 , z2;
287

```

```

288         u1= static_cast<double>(rand()) ;    u2=static_cast<double>(rand()) ;
289
290         u1 = u1 / (RAND_MAX*1.0) ; u2 = u2 / (RAND_MAX*1.0) ;
291
292         r=sqrt(-2*log(u1)) ; th = 2*pi*u2 ;
293         z1 = scalar*r*cos(th) ;
294
295
296         z2 = scalar*r*sin(th) ;
297
298         y[ii].re += z1+real ;
299
300         y[ii].im += z2+imag ;
301
302
303     }
304
305     return ; }
306
307 /** ilopoihsi ths apodiamorfosis QAM */
308 void qamdemod(cmplx y[m] , int x[m]) {
309
310     /*
311
312         k = sqrt(m) ;
313         r = 2*(0:k-1) - k + 1 ;
314         [xi, yi] = meshgrid(r) ;
315         c = xi + 1j*flipud(yi) ;
316         c = c(:) ;
317     */
318     // dimiourgeitai to pli8os ton stoixeion
319     cmplx c[q] ;
320     int ii , jj , kk = 0 ;
321     for(ii=1-k ; ii<k ; ii+=2)
322         for(jj=k-1 ; jj+k>0 ; jj-=2) {
323             c[kk].re = ii ;
324             c[kk].im = jj ;
325             kk++ ; }
326
327     *****
328     ***** test c[q] construction *****
329     /*
330         for(kk=0 ; kk<q ; kk++) {
331             cout << c[kk].re << "+i " << c[kk].im << endl ; }
332         cout << endl ;
333     */
334     *****
335
336     /*
337
338         z = zeros(size(y)) ;
339         for k = 1:length(y)
340             [nil ind] = min(abs(y(k) - c)) ;
341             z(k) = ind - 1 ;
342         end
343     */
344
345     double mn ;
346     int ind ;

```

```

347   cmplx z ;
348   // eyresi toy elaxistou migadikou
349   for(kk=0 ; kk<m ; kk++) {
350       z.re = y[kk].re - c[0].re ;
351       z.im = y[kk].im - c[0].im ;
352       mn = modul(z) ;
353       ind = 0 ;
354   // entopismos tis akrivis 8esis tou stoixeiou
355       for(ii=1 ; ii<q ; ii++) {
356           z.re = y[kk].re - c[ii].re ;
357           z.im = y[kk].im - c[ii].im ;
358           if (modul(z)<mn+eps) {
359               mn = modul(z) ;
360               ind = ii ; } }
361       x[kk] = ind ; }
362
363   return ; }
364
365   int main(){
366   int i, ebn0 ;
367   cmplx data_mod[m] , data_ifft[m] , data_guard[dim] , data_rem_all_guard[m] ;
368   double data[2*m+1] ;
369
370
371   /**/ arxikopoiisi tixaias akolouthias /**/
372
373   srand(time(NULL)) ;
374
375   /**/ arxikopoiisi ton pinakon snr kai rate0 /**/
376
377   for(i=0 ; i<lsnr ; i++) {
378       snr[i] = i ; rate0[i] = 0 ; }
379
380
381   for(ebn0=0 ; ebn0<lsnr ; ebn0++) {
382
383       construct_data_in() ;
384
385       /* Data_In = randint(m,1,q) */
386       /****** OBSOLETE (previous version) *****/
387       /* for(i=0 ; i<m ; i++) data_in[i] = rand()%q ; */
388       /******
389
390       /****** test Data_In = randint(m,1,q) *****/
391       /******
392       /*
393       for(i=0 ; i<m ; i++) cout << x[i] << " " ; cout << endl ;
394       */
395       /******
396
397       qammod(data_in , data_mod) ; //kaleitai h qammod
398
399       /**/ etoimazontai ta dedomena gia ton ipologismo tou ifft /**/
400       for(i=0 ; i<m ; i++) {
401           data[2*i+1] = data_mod[i].re ;
402           data[2*i+2] = data_mod[i].im ; }
403
404       //ypologizetai o IFFT
405       fft(data , m , -1) ;

```

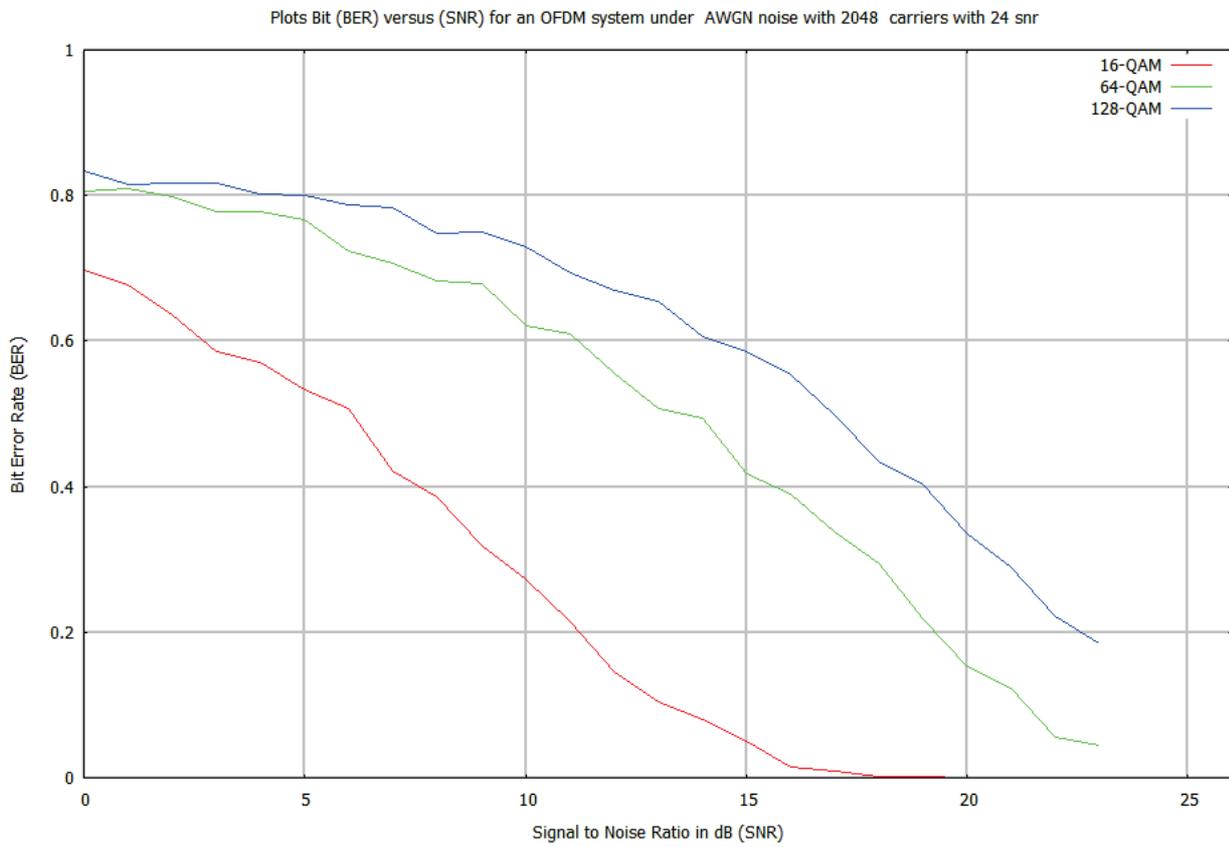
```

406
407
408     for(i=0 ; i<m ; i++) {
409         data_ifft[(m-i)%m].re = data[2*i+1]/m ;
410         data_ifft[(m-i)%m].im = data[2*i+2]/m ; }
411
412     /** eisagigi Guard Interval ***/
413     for(i=0 ; i<m*gi ; i++) {
414         data_guard[i].re = data_ifft[m-m*gi+i].re ;
415         data_guard[i].im = data_ifft[m-m*gi+i].im ; }
416     for(i=m*gi ; i<m+m*gi ; i++) {
417         data_guard[i].re = data_ifft[i-m*gi].re ;
418         data_guard[i].im = data_ifft[i-m*gi].im ; }
419
420     /* Data_Noise = awgn(Data_Tx,ebn0,'measured'); */
421     add_white_noise(data_guard , ebn0) ;
422
423     /*
424         Data_Rx=reshape(Data_Noise,a,b);
425         Data_Removeal_Gaurd = Data_Rx(Guard_Interval+1:M+Guard_Interval) ;
426     */
427     /** afairesi tou Guard Interval ***/
428     for(i=0 ; i<m ; i++) {
429         data_rem_all_guard[i].re = data_guard[i+m*gi].re ;
430         data_rem_all_guard[i].im = data_guard[i+m*gi].im ; }
431
432     ***** ypologismos tou fft *****
433
434     /** prepare data for the fft calculation ***/
435     for(i=0 ; i<m ; i++) {
436         data[2*i+1] = m*data_rem_all_guard[i].re ;
437         data[2*i+2] = m*data_rem_all_guard[i].im ; }
438
439     fft(data , m , 1) ;
440
441     /** diavazontai oi times pou prokiptoun apo ton FFT ***/
442     for(i=0 ; i<m ; i++) {
443         data_rem_all_guard[(m-i)%m].re = data[2*i+1]/m ;
444         data_rem_all_guard[(m-i)%m].im = data[2*i+2]/m ; }
445
446     qamdemod(data_rem_all_guard , data_out) ;
447
448
449     int diff_num=0 ;
450     for(i=0 ; i<m ; i++)
451         if (data_in[i]!=data_out[i]) diff_num ++ ; // sigrisi data_in
452     kai data_out gia ypologismo diaforon
453
454
455     rate0[ebn0] =static_cast<double>(diff_num*1.0 / (m*1.0) ) ; } //
456     ypologismos tou BER
457
458
459
460
461
462     ***** exagogi ton dedomenon se arxeio txt *****
463
464     ofstream fout ("co_rayl.txt");

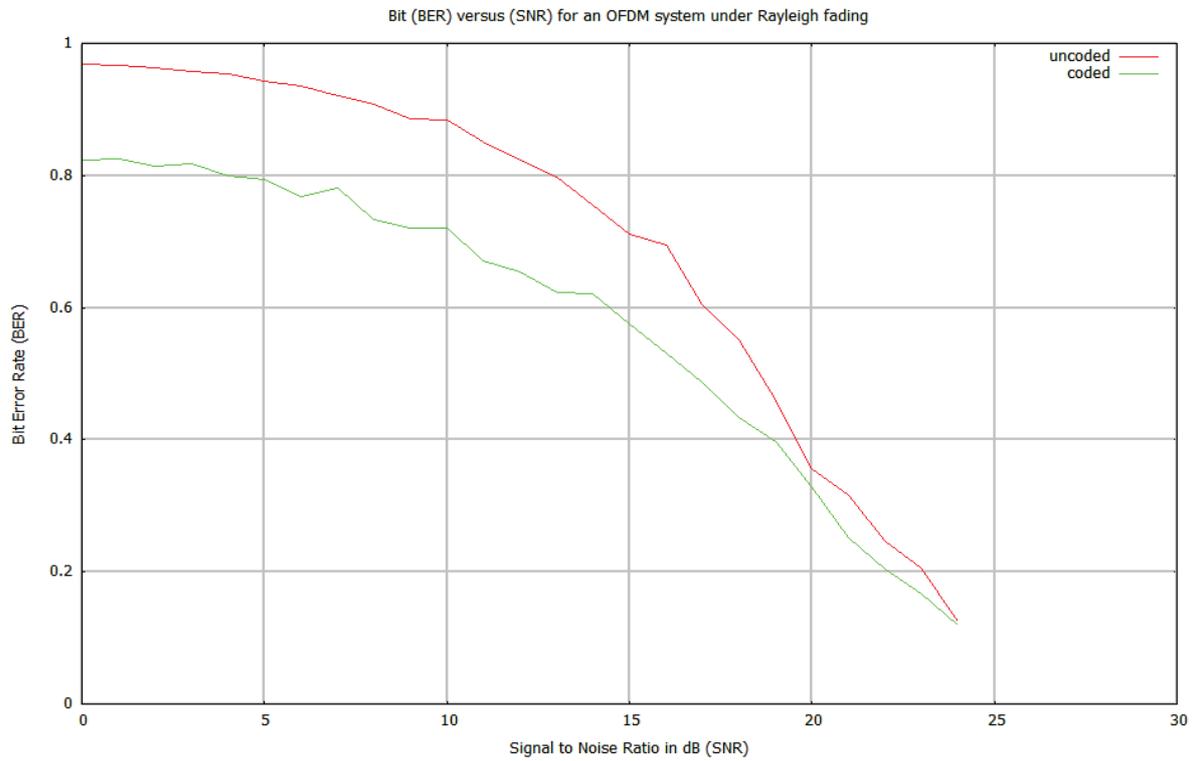
```

```
465
466 for(i=0 ; i<lsnr ; i++)
467     fout << rate0[i] << endl ; // ypologizontai ta simeia tis grafikis
468 parastasis
469     cout << double( clock() - startTime ) / (double)CLOCKS_PER_SEC<< "
470 seconds." << endl; //ypologismos toy xronou ektelesis toy programmatos
471 fout.close();
472
473
474
475 /***** grafiki anaparastasi ton dedomenon me to gnuplot *****/
476
477     requirements:
478         gnuplot must be installed in the system
479         gnuplot is an open source graph utility compatible with c/c++
480         fully documented and can be downloaded from
481             http://www.gnuplot.info/
482         the path of gnuplot must be in the PATH enviroment
483         (there is an option for automatic update during installation)
484 */
485
486 FILE *gnuplot = popen("gnuplot", "w");
487 fprintf(gnuplot,"set title 'Plots Bit (BER) versus (SNR) for an OFDM system
488 under AWGN noise'\n");
489 fprintf(gnuplot,"set xlabel 'Signal to Noise Ratio in dB (SNR)'\n");
490 fprintf(gnuplot,"set ylabel 'Bit Error Rate (BER)'\n");
491 fprintf(gnuplot, "set autoscale\n");
492 fprintf(gnuplot,"set style function lines\n");
493 fprintf(gnuplot, "set grid\n");
494 fprintf(gnuplot, "plot '-' with lines\n");
495 for (i=0 ; i<lsnr ; i++)
496     fprintf(gnuplot, "%d %f \n", i , rate0[i]);
497
498 fprintf(gnuplot, "e\n");
499 fflush(gnuplot);
500
501 int vv ; cin >> vv ;
502 return 0 ; }
```

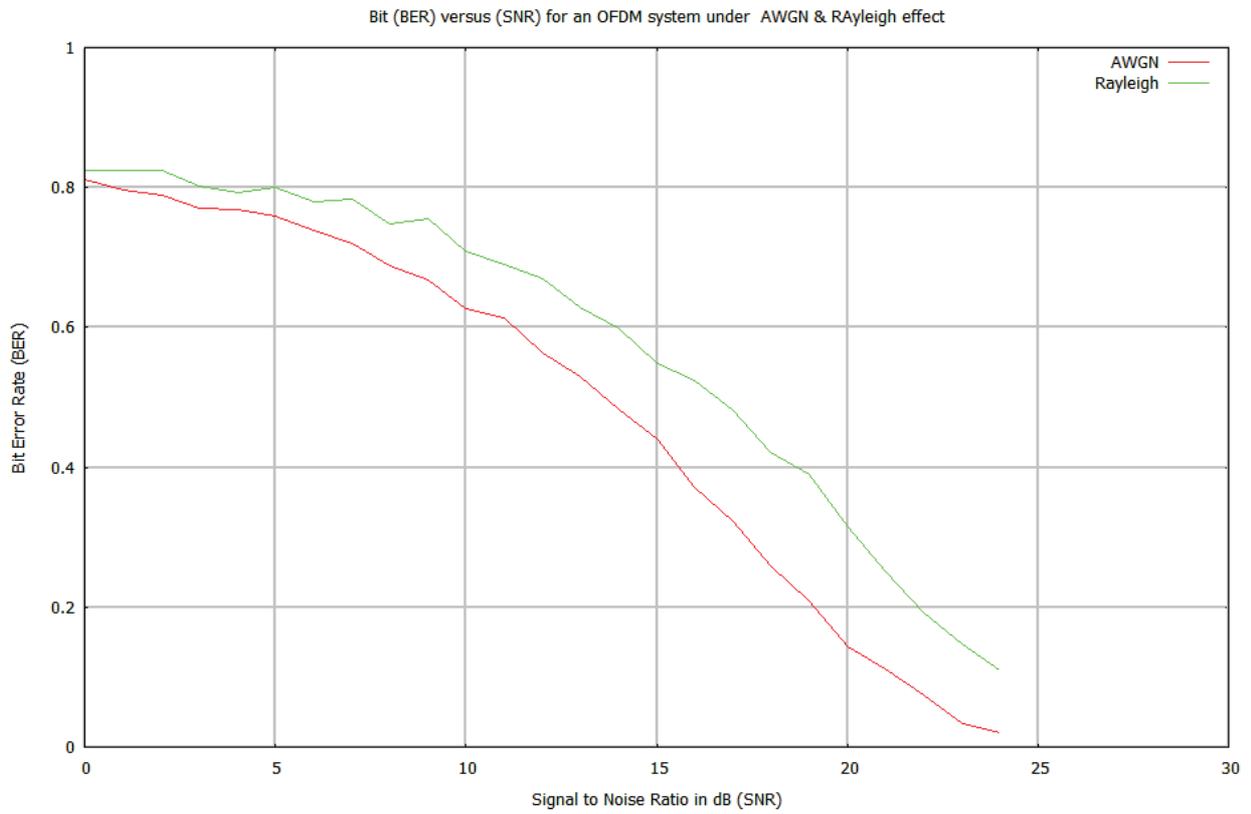
ΚΕΦΑΛΑΙΟ 4^ο ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΗΣ



Όσο αυξάνεται η τάξη της διαμόρφωσης QAM τόσο μεγαλώνει το BER.



Παρατηρούμε πως το BER χαμηλώνει όταν στο σύστημα χρησιμοποιείται συνελκτική κωδικοποίηση.



Παρατηρούμε πως το BER αυξάνεται σε περιβάλλον πολυδιαδρομικής διάδοσης.

Παράρτημα Α: Παράμετροι OFDM

Παράμετρος	Λειτουργία 8K	Λειτουργία 2K
Αριθμός Φερόντων	6 817	1 705
Τιμή του Φέροντος με αριθμό K_{\min}	0	0
Τιμή του Φέροντος με αριθμό K_{\max}	6816	1704
Διάρκεια Συμβόλου T_U	896 μ s	224 μ s
Απόσταση Φερόντων $1/T_U$	1 116 Hz	4 464 Hz
Απόσταση μεταξύ φερόντων K_{\min} και K_{\max} $(K-1)1/T_U$	7,61 MHz	7,61MHz

Πίνακ. 4 Τιμές παραμέτρων καναλιού 8MHz για τους 2k και 8k τρόπους λειτουργίας

Οι τιμές για διάφορες χρονικές παραμέτρους δίνονται σε πολλαπλάσια της βασικής περιόδου T και σε μ s. Η βασική περίοδος T είναι $7/64 \mu$ s για κανάλια των 8MHz, $1/8 \mu$ s για κανάλια των 7 MHz και $7/48 \mu$ s για κανάλια των 6MHz. Το εκπεμπόμενο σήμα περιγράφεται από την παρακάτω σχέση:

$$s(t) = \text{Re} \left\{ e^{j2\pi f_c t} \sum_{m=0}^{\infty} \sum_{l=0}^{67} \sum_{k=K_{\min}}^{K_{\max}} c_{m,l,k} \times \psi_{m,l,k}(t) \right\}$$

$$\psi_{m,l,k}(t) = \begin{cases} e^{j2\pi \frac{k'}{T_U}(t - \Delta - l \times T_s - 68 \times m \times T_s)} & (l + 68 \times m) \times T_s \leq t \leq (l + 68 \times m + 1) \times T_s \\ 0 & \text{αλλού} \end{cases}$$

όπου

k δηλώνει τον αριθμό φέροντος

l δηλώνει τον αριθμό συμβόλου OFDM

m δηλώνει τον αριθμό του πλαισίου μετάδοσης

K είναι ο αριθμός των μεταδιδόμενων φερόντων

T_s είναι η διάρκεια συμβόλου (συνολική μαζί με το διάστημα «φρουρό»)

T_U είναι το ανάστροφο του διαστήματος φερόντων

Δ είναι η διάρκεια του διαστήματος φρουρού

f_c είναι η κεντρική συχνότητα του σήματος RF

k' είναι ο δείκτης φέροντος σχετικός με την κεντρική συχνότητα:

$$k' = k - (K_{\max} + K_{\min}) / 2$$

$c_{m,0,k}$ μιγαδικό σύμβολο για το φέρον k του συμβόλου δεδομένου αρ.1 στο πλαίσιο m

$c_{m,1,k}$ μιγαδικό σύμβολο για το φέρον k του συμβόλου δεδομένου αρ.2 στο πλαίσιο m

$c_{m,67,k}$ μιγαδικό σύμβολο για το φέρον k του συμβόλου δεδομένου αρ.68 στο πλαίσιο m

Λειτουργία	Λειτουργία 8K				Λειτουργία 2K			
Διάστημα Ασφάλειας ΔΕ/ TU	½	1/8	1/16	1/32	½	1/8	1/16	1/32
Διάρκεια Συμβόλου εκτός TU	8 192 × T 896 μs (Σημείωση)				2 048 × T 224 μs (Σημείωση)			
Διάρκεια Διαστήματος Ασφάλειας Δ	2 048 × T 224 μs	1 024 × T 112 μs	512 × T 56 μs	256 × T 28 μs	512 × T 56 μs	256 × T 28 μs	128 × T 14 μs	64 × T 7 μs
Διάρκεια Συμβόλου TS= Δ+ TU	10 240 × T 1120 μs	9 216 × T 1008 μs	8 704 × T 952 μs	8 448 × T 924 μs	2 560 × T 280 μs	2 304 × T 252 μs	2 176 × T 238 μs	2 112 × T 231 μs

Πίνακ. 4 Τιμές για κανάλια των 8 MHz

Παράρτημα Β: Μορφή μετάδοσης ενός TPS

Ένα μπλοκ TPS αποτελείται συνολικά από 68 bits: 1 bit για αρχικοποίηση, 16 bit συγχρονισμού, 37 bits πληροφορίας, 14 bits πλεονασμού για προστασία από σφάλματα. Από τα 37 bits χρησιμοποιούνται μόνο τα 31 τα υπόλοιπα 6 θέτονται "0".

Το πρώτο bit s_0 χρησιμοποιείται για αρχικοποίηση. Τα bit από s_1 έως το s_{16} αποτελούν μια λέξη συγχρονισμού. Τα bits από 1 έως το 16 αποτελούν μια λέξη συγχρονισμού.

Το πρώτο και τρίτο TPS μπλόκ σε κάθε υπερπλαίσιο έχουν την ακόλουθη λέξη συγχρονισμού $s_{1-16} = 0011010111101110$.

Ενώ, το δεύτερο και τέταρτο TPS μπλόκ σε κάθε υπερπλαίσιο έχουν την ακόλουθη λέξη συγχρονισμού $s_{1-16} = 1100101000010001$.

Τα πρώτα 6 bits πληροφορίας από το s_{17} έως το s_{22} (από τα 37) χρησιμοποιούνται ως δείκτης διάρκειας TPS ώστε να σηματοδοθεί ο αριθμός των χρησιμοποιούμενων bits του TPS. Η μετάδοση της ταυτοποίησης κελιών είναι προαιρετική. Ο δείκτης διάρκειας TPS παίρνει τις εξής τιμές:

010111 όταν η πληροφορία ταυτοποίησης κελιού δε μεταδίδεται (χρήση 23 TPS bits).

011111 όταν η πληροφορία ταυτοποίησης κελιού μεταδίδεται (χρήση 23 TPS bits).

Τα πλαίσια που αποτελούν το υπερπλαίσιο αριθμούνται από το 1 έως το 4. Η αντιστοίχιση τους στα bits από το $s_{23} - s_{24}$ δίνεται στον παρακάτω πίνακα.

Bits s_{23}, s_{24}	Αριθμός πλαισίου
0 0	Αριθμός πλαισίου 1 στο υπερπλαίσιο
0 1	Αριθμός πλαισίου 2 στο υπερπλαίσιο

1 0	Αριθμός πλαισίου 3 στο υπερπλαίσιο
1 1	Αριθμός πλαισίου 4 στο υπερπλαίσιο

Ο αστερισμός σηματοδοτείται απο 2 bits σύμφωνα με τον ακόλουθο πίνακα.

Bits s_{25}, s_{26}	Χαρακτηριστικά αστερισμού
0 0	QPSK
0 1	16-QAM
1 0	4-QAM
1 1	Κατοχυρωμένο

Οι πληροφορίες ιεραρχίας καθορίζουν αν η μετάδοση είναι ιεραρχική ή όχι , όπως επίσης και την τιμή του α , η ποία σηματοδοτείται απο 3 bits όπως φαίνεται στον πίνακα που ακολουθεί.

Bits s_{27}, s_{28}, s_{29}	Τιμή α
0 0 0	Μη ιεραρχική
0 0 1	$\alpha = 1$
0 1 0	$\alpha = 2$
0 1 1	$\alpha = 4$

Δυο διαφορετικούς ρυθμούς κωδικοποίησης μπορούν να εφαρμοστούν σε δυο διαφορετικά επίπεδα διαμόρφωσης με στόχο την επίτευξη της ιεραρχίας. Η μετάδοση ξεκινά με το ρυθμό κωδικοποίησης για το επίπεδο υψηλής προτεραιότητας της διαμόρφωσης HP και τελειώνει με αυτο της χαμηλής προτεραιότητας LP. Οι ρυθμοί κωδικοποίησης σηματοδοτούνται σύμφωνα με τον επόμενο πίνακα.

Bits s_{30}, s_{31}, s_{32} (HP ροή) s_{33}, s_{34}, s_{35} (LP ροή)	Ρυθμός κωδικοποίησης
0 0 0	$\frac{1}{2}$
0 0 1	$\frac{2}{3}$
0 1 0	$\frac{3}{4}$
0 1 1	$\frac{5}{6}$
1 0 0	$\frac{7}{8}$
1 0 1	Κατοχυρωμένο
1 1 0	Κατοχυρωμένο
1 1 1	Κατοχυρωμένο

Οι τιμές του διαστήματος ασφαλείας σηματοδοτούνται ως εξής:

Bits s_{36}, s_{37}	Τιμές διαστήματος ασφαλείας (Δ/TU)
0 0	1/32
0 1	1/16
1 0	1/8
1 1	1/4

Δυο bits χρησιμοποιούνται για τη σηματοδότηση της λειτουργίας (8K ή 2K)

Bits s_{38}, s_{39}	Λειτουργία μετάδοσης
0 0	2K
0 1	8K
1 0	DVB-H
1 1	Κατοχυρωμένο

Τα 8 bits $s_{40} - s_{47}$ χρησιμοποιούνται για να αναγνωρισθεί το κελί από το οποίο προέρχεται το σήμα. Το πιο σημαντικό byte του cell_id για παράδειγμα $b_{15} - b_8$ θα μεταδοθεί σε ένα υπερπλαίσιο με αριθμό πλαισίου 1 και 3. Τα λιγότερο σημαντικό byte του cell_id για παράδειγμα $b_{15} - b_8$, θα μεταδοθεί σε ένα υπερπλαίσιο με αριθμό πλαισίου 2 και 4. Η αντίστοιχη των bits φαίνεται παρακάτω.

Αριθμός bit TPS	Αριθμός πλαισίου 1 ή 3	Αριθμός πλαισίου 2 ή 4
s_{40}	cell_id b_{15}	cell_id b_7
s_{41}	cell_id b_{14}	cell_id b_6
s_{42}	cell_id b_{13}	cell_id b_5
s_{43}	cell_id b_{12}	cell_id b_4
s_{44}	cell_id b_{11}	cell_id b_3
s_{45}	cell_id b_{10}	cell_id b_2
s_{46}	cell_id b_9	cell_id b_1
s_{47}	cell_id b_8	cell_id b_0

Τα bits από s_{48} έως s_{53} χρησιμοποιούνται στο DVB_H. Στα 53 bits συγχρονισμού και πληροφορίας προστίθενται και 14 bits (s_{54} to s_{67}) ισοτιμίας του συντομευμένου κώδικα BCH (67,53, $t = 2$).

Το πολυώνυμο γεννήτρια είναι: $h(x) = x^{14} + x^9 + x^8 + x^6 + x^5 + x^4 + x^2 + x + 1$

Οι πληροφορίες αυτές στέλνονται αναφορικά στο επόμενο πλαίσιο από αυτό που τις μεταφέρει ώστε να ενημερώνεται ο δέκτης για τυχόν αλλαγές στις παραπάνω παραμέτρους, δεδομένου ότι τις αρχικές τους τιμές τις γνωρίζει.

Τα πιλοτικά σύμβολα διαμορφώνονται με πληροφορίες αναφοράς, η τιμή που εκπέμπεται για αυτά τα σύμβολα είναι γνωστή στο δέκτη και το επίπεδο ενέργειας τους είναι μεγαλύτερο από αυτό των σημάτων - δεδομένων. Ο αριθμός των χρήσιμων φερόντων δεδομένων είναι 1512 για 2K OFDM και 6048 για 8K OFDM. Τα συνεχή κελιά-πιλότοι, τοποθετούνται στο ίδιο υπό-κανάλι για όλα τα OFDM σύμβολα. Χρησιμοποιούνται 177 συνεχή σύμβολα-πιλότοι στην 8k λειτουργία και 45 για την 2k λειτουργία, τοποθετημένα σε θέσεις που ορίζονται στο παρακάτω από το παρακάτω σχήμα.

Code rate	QPSK		16-QAM		64-QAM	
	2K mode	8K mode	2K mode	8K mode	2K mode	8K mode
1/2	252	1 008	504	2 016	756	3 024
2/3	336	1 344	672	2 688	1 008	4 032
3/4	378	1 512	756	3 024	1 134	4 536
5/6	420	1 680	840	3 360	1 260	5 040
7/8	441	1 764	882	3 528	1 323	5 292

Σχ.25 Αριθμός πακέτων RS για κάθε OFDM υπερπλαίσιο για όλους

τους συνδιασμούς των ρυθμών κωδικοποίησης και διαμόρφωσης

Η δομή του πλαισίου OFDM επιτρέπει για έναν ακέραιο αριθμό πακέτων μεγέθους 204 bytes του κώδικα Reed-Solomon να μεταδίδονται σε ένα υπερπλαίσιο χωρίς να μας ενδιαφέρει ο αστερισμός, η διάρκεια διαστήματος ασφαλείας, ο ρυθμός κωδικοποίησης το εύρος ζώνης του καναλιού.

Παράρτημα Γ: Τεχνικά χαρακτηριστικά DVB-T

Παρακάτω συνοψίζονται τα τεχνικά χαρακτηριστικά του προτύπου DVB-T:

Κωδικοποίηση	MPEG-2
Ζώνες μετάδοσης	UHF(470-862 MHz) και VHF (174-230 MHz)
Εύρος ζώνης καναλιών	6,7 ή 8 MHz
Επιλογές διαμόρφωσης	QPSK,16-QAM,64-QAM σε σχήμα. πολλαπλών ορθογώνιων φερόντων (Coded OFDM).
Ρυθμοί FEC (Forward Error Correction)	1/2 ,2/3, 3/4, 5/6, 7/8
Επιλογές Guard Interval	1/4, 1/8, 1/16, 1/32
Concatenated Coding, Convolutional + Block Coding (Κωδικοποίηση δυο επιπέδων, συνελκτική + μπλοκ κωδικοποίηση)	
Ωφέλιμο bitrate	4.98 Mbps έως 31.67 Mbps

Συντομογραφίες

BER	Bit Error Ratio
DFT	Discrete Fourier Transform
DVB	Digital Video Broadcastin
DVB-H	DVB-Handheld
DVB-T	DVB-Terrestrial
FIFO	First-In, First-Out shift register
FFT	Fast Fourier Transform
HDTV	High Definition TeleVision
HP	High Priority bit stream
LP	Low Priority bit stream
MPEG	Moving Picture Experts Group
MSB	Most Significant Bit
MUX	Multiplex
OFDM	Orthogonal Frequency Division Multiplexing
PRBS	Pseudo-Random Binary Sequence
QAM	Quadrature Amplitude Modulation
QPSK	Quaternary Phase Shift Keying
RF	Radio Frequency
RS	Reed-Solomon
s_i	TPS bit index
SDTV	Standard Definition TeleVision
SFN	Single Frequency Network
TPS	Transmission Parameter Signalling
TV	Television
UHF	Ultra-High Frequency
VHF	Very-High Frequency

Βιβλιογραφία

1. ETSI EN 300 744 v1.6.1 (2009-01), European Standard (Telecommunications series) “Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television”
2. Understanding Digital Terrestrial Broadcasting, Seamus O.Leary Artech House (2000).
3. <http://igorfun.com/dvb-t/>
4. <http://en.wikipedia.org/wiki/DVB-T>
5. http://www.digitaltvinfo.gr/index.php?option=com_magazine&task=viewArticle&id=1135&Itemid=102
6. <http://www.dvb.org/>
7. http://code.ua.pt/projects/bookco1011/wiki/_973_How_OFDM_Works
8. <http://flylib.com/books/en/4.136.1.34/1/>