



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υλοποίηση του φυσικού επιπέδου του πομπού του
πρωτοκόλλου IEEE 802.11a σε υλικο

Αργυρίου Αλέξανδρος

Πτυχιακή Εργασία

Υπέθυνος

Σταμούλης Γεώργιος

Συνυπεύθυνος

Άχμεντ Μάχντι

Λαμία, 2017



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υλοποίηση του φυσικού επιπέδου του πομπού του
πρωτοκόλλου IEEE 802.11a σε υλικο

Αργυρίου Αλέξανδρος

Πτυχιακή Εργασία

Υπέθυνος

Σταμούλης Γεώργιος

Συνυπέθυνος

Άχμεντ Μάχντι

Λαμία, 2017

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.

2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία:/...../20.....

Ο – Η Δηλ.

(Υπογραφή)

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κ. Ahmed Mahdi, που βοήθησε για την εκπόνηση αυτής της εργασίας. Ήταν πάντα πρόθυμος να με συμβουλέψει και να με καθοδηγήσει και πάντα έδειχνε κατανόηση στις δυσκολίες που αντιμετώπισα. Τον ευχαριστώ και για την επιλογή του θέματος της εργασίας, γιατί οι γνώσεις που αποκόμισα από αυτήν ήταν πολλές και πολύτιμες.

Περίληψη

Στην παρούσα εργασία καλούμαστε να υλοποιήσουμε το φυσικό επίπεδο του πομπού του πρωτοκόλλου IEEE 802.11a. Το πρωτόκολλο αυτό αναφέρεται σε ασύρματα τηλεπικοινωνιακά συστήματα. Για να κατανοηθεί το πρωτόκολλο και η υλοποίησή του, πρώτα θα γίνει μία εισαγωγή στα επικοινωνιακά συστήματα, μετά στα τηλεπικοινωνιακά και ασύρματα συστήματα και τέλος θα περιγραφεί συγκεκριμένα η μέθοδος που χρησιμοποιεί το πρωτόκολλο για την μετάδοση της πληροφορίας. Επόμενο βήμα είναι η πλήρης αναφορά της δομής του φυσικού επιπέδου του πρωτοκόλλου και η αναλυτική αναφορά του σκοπού της χρήσης και της λειτουργίας των υποσυστημάτων του επιπέδου του πρωτοκόλλου που μας ενδιαφέρει στην εργασία αυτή. Το επίπεδο αυτό είναι το PLCP. Ενώ τα υποσυστήματα που περιέχονται στο επίπεδο PLCP είναι ο scrambler, ο συνελκτικός κωδικοποιητής, ο puncture, ο interleaver, ο διαμορφωτής, ο symbol former, το σύστημα πιλοτικών τιμών, ο αντίστροφος γρήγορος μετασχηματισμός Fourier, η εισαγωγή κυκλικού προθέματος και η παραθύρωση.

Στην συνέχεια εξηγούμε τι είναι ένα FPGA και ποια είναι η ροή σχεδίασης σε αυτό. Ενώ δόθηκε και μία σειρά από στιγμιότυπα ολόκληρης της διαδικασίας υλοποίησης της σχεδίασης με τη βοήθεια του εργαλείου Vivado της Xilinx. Αυτό το εργαλείο θα συνθέσει και θα εγκαταστήσει την σχεδίαση στο FPGA. Εφόσον εξηγηθεί τι ακριβώς είναι ένα FPGA και πως χρησιμοποιούνται τα εργαλεία σχεδίασης, δίνεται η αναλυτική περιγραφή των κυκλωμάτων του φυσικού επιπέδου του πομπού του πρωτοκόλλου IEEE 802.11a. Οι σχεδιάσεις αυτές γράφτηκαν στην γλώσσα περιγραφής υλικού VHDL. Η σχεδίαση των κυκλωμάτων του φυσικού επιπέδου του πρωτοκόλλου ήταν ο βασικός στόχος της εργασίας αυτής. Μετά την ολοκλήρωση της σχεδίασης χρειάστηκε να παρουσιαστεί και ο τρόπος με τον οποίο το κάθε ένα από αυτά τα κυκλώματα προσομοιώθηκε και επαληθεύτηκε η σωστή λειτουργία του. Για τον σκοπό αυτό δόθηκε η αναλυτική ροή που ακολουθήθηκε για την προσομοίωση και την επαλήθευση του κυκλώματος του scrambler. Για κάθε ένα από τα κυκλώματα που σχεδιάστηκαν ακολουθήθηκε η ίδια μέθοδος προσομοίωσης και επαλήθευσης. Με όλα τα παραπάνω η παρούσα εργασία καλύπτει τον σκοπό της.

Λέξεις κλειδιά: IEEE 802.11a, πομπός, φυσικό επίπεδο, πρωτόκολλο επικοινωνίας, ασύρματο τηλεπικοινωνιακό σύστημα, OFDM, FPGA, VHDL, WiFi

Περιεχόμενα

Ευχαριστίες.....	5
Περίληψη.....	6
Περιεχόμενα.....	7
1 Εισαγωγή.....	9
1.1 Τηλεπικοινωνιακό σύστημα.....	9
1.2 Ασύρματο τηλεπικοινωνιακό σύστημα.....	10
1.3 Ορθογώνια πολυπλεξία με διαίρεση συχνότητας.....	10
1.4 Πρωτόκολλα επικοινωνίας.....	11
1.3.1 Το μοντέλο OSI.....	11
2 Το πρωτόκολλο επικοινωνίας IEEE 802.11a.....	14
2.1 Περιγραφή πεδίων του PLCP πακέτου	15
2.1.1 Πεδίο preamble.....	15
2.1.2 Πεδίο signal.....	16
2.1.3 Πεδίο data.....	17
2.2 Ροή δεδομένων του επιπέδου PLCP.....	18
2.3 Τα συστήματα του επιπέδου PLCP.....	19
2.3.1 Scrambler.....	20
2.3.2 Συνελκτικός κωδικοποιητής.....	21
2.3.3 Puncture.....	23
2.3.4 Interleaver.....	25
2.3.5 Σύστημα διαμόρφωσης.....	27
2.3.6 Σύστημα συνένωσης του συμβόλου.....	30
2.3.7 Σύστημα παραγωγής πιλοτικών τιμών.....	30
2.3.8 Αντίστροφος γρήγορος μετασχηματισμός Fourier.....	31
2.3.9 Εισαγωγή τιμών φύλαξης και παραθύρωσης.....	32
3 FPGA.....	33
3.1 Ροή σχεδίασης στο FPGA.....	36
3.2 Ροή σχεδίασης στο Vivado της Xilinx.....	38
3.2.1 Δημιουργία project και εισαγωγή αρχείων σχεδίασης.....	38
3.2.2 Σύνθεση σχεδίασης.....	41
3.2.3 Δημιουργία του αρχείου περιορισμού.....	42
3.2.4 Επεξεργασία του αρχείου περιορισμού.....	44
3.3 Αναφορές σχεδίασης.....	48
4 Υλοποίηση του φυσικού επιπέδου του πομπού του πρωτοκόλλου IEEE 802.11a.....	50
4.1 Ανώτερο επίπεδο σχεδίασης του επιπέδου PLCP.....	50
4.2 Μηχανή πεπερασμένων καταστάσεων.....	53

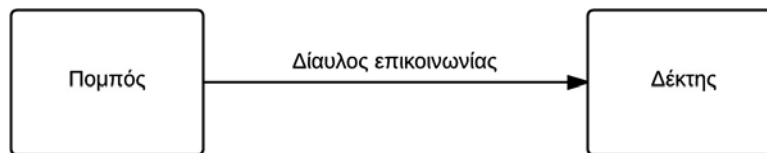
4.3 Προοίμιο.....	53
4.4 Κύκλωμα συνένωσης του πεδίου signal.....	54
4.5 Κύκλωμα συνένωσης του πεδίου data.....	56
4.6 Scrambler.....	57
4.7 Συνελκτικός κωδικοποιητής.....	58
4.8 Puncture.....	59
4.9 Interleaver.....	60
4.10 Κύκλωμα διαμόρφωσης.....	64
4.11 Κύκλωμα συνένωσης του συμβόλου.....	66
4.12 Κύκλωμα παραγωγής πιλοτικών τιμών.....	69
4.13 Αντίστροφος γρήγορος μετασχηματισμός Fourier.....	70
4.14 Εισαγωγή τιμών φύλαξης και παραθύρωσης.....	74
5 Προσομοίωση και επαλήθευση της σχεδίασης.....	76
6 Συμπεράσματα.....	83
7 Μελλοντικές εργασίες.....	83
8 Παραρτήματα.....	84
8.1 Αναπαράσταση ως προς 2 αριθμών με σταθερή υποδιαστολή.....	84
8.2 Αναπαράσταση μιγαδικών αριθμών.....	84
8.3 Πράξεις με μιγαδικούς αριθμούς και ικανοποίηση του ζητούμενου κρίσιμου μονοπατιού.....	85
9 Αναφορές.....	86

1 Εισαγωγή

Η εργασία αυτή αποσκοπεί στην περιγραφή, στην σχεδίαση και στην υλοποίηση των ψηφιακών κυκλωμάτων που απαρτίζουν το φυσικό επίπεδο του πομπό του πρωτοκόλλου IEEE 802.11a [1]. Όμως πριν περιγραφεί ο πομπός και τα υποσυστήματά του, θα πρέπει να αναφερθούν κάποιοι γενικοί ορισμοί γύρω από τα τηλεπικοινωνιακά συστήματα, οι οποίοι θα βοηθήσουν στην καλύτερη κατανόηση του συστήματος του πομπό. Έτσι στο εισαγωγικό αυτό κεφάλαιο θα αναφέρουν οι έννοιες που πρέπει να γνωρίζει κανείς για να κατανοήσει ευκολότερα το υπόλοιπο περιεχόμενο της εργασίας.

1.1 Τηλεπικοινωνιακό σύστημα

Επικοινωνία είναι η διαδικασία κατά την οποία μεταδίδεται πληροφορία από ένα σημείο σε ένα άλλο. Για παράδειγμα, τέτοιου είδους επικοινωνία είναι η ομιλία μεταξύ δύο ανθρώπων. Υπάρχουν διάφορα είδη επικοινωνίας, όμως αυτό με το οποίο ασχολείται η παρούσα εργασία είναι η τηλεπικοινωνία. Η τηλεπικοινωνία είναι η μορφή της επικοινωνίας που επιτυγχάνεται μεταξύ δύο οποιονδήποτε σημείων Α και Β, τα οποία χωρίζονται από μεγάλη απόσταση. Ενώ τηλεπικοινωνιακό σύστημα είναι το σύστημα που καθιστά δυνατή την επικοινωνία μεταξύ των σημείων αυτών. Η δομή των τηλεπικοινωνιακών συστημάτων αποτελείται από τρία βασικά τμήματα. Η δομή αυτή απεικονίζεται στο σχήμα 1 και όπως φαίνεται αποτελείται από τον πομπό, τον δέκτη και τον δίαυλο επικοινωνίας.



Σχήμα 1. Μοντέλο επικοινωνιακού συστήματος.

Οι επικοινωνίες μεταξύ δύο σημείων που χωρίζονται από μεγάλη απόσταση μας υποχρεώνουν να αλλάζουμε την μορφή της πληροφορίας πριν την μεταδώσουμε, ώστε να αποσταλεί αποδοτικά στον δέκτη του συστήματος. Η αλλαγή της μορφής της πληροφορίας καθορίζεται από μία σειρά κανόνων. Τα σύνολα των κανόνων αυτών απαρτίζουν τα διάφορα πρωτόκολλα επικοινωνίας. Τα πρωτόκολλα επικοινωνίας περιγράφονται στην ενότητα 1.4. Ενώ, συγκεκριμένα, οι κανόνες που ορίζει το πρωτόκολλο IEEE 802.11a αναφέρονται στην ενότητα 2.

Ο πομπός του συστήματος επικοινωνίας είναι αυτός που επεξεργάζεται την πληροφορία σύμφωνα με το επιθυμητό πρωτόκολλο και τελικά την μεταδίδει στον δέκτη μέσω του διαύλου επικοινωνίας. Ο δίαυλος επικοινωνίας είναι το φυσικό μέσο που συνδέει τον πομπό με τον δέκτη. Για παράδειγμα ο δίαυλος επικοινωνίας μπορεί να είναι ένα καλώδιο χαλκού, ένα καλώδιο οπτικής ίνας κ.τ.λ. Από την άλλη πλευρά ο δέκτης λαμβάνει και επεξεργάζεται το σήμα πληροφορίας. Η επεξεργασία που λαμβάνει χώρα στην πλευρά του δέκτη είναι ακριβώς η αντίστροφη διαδικασία από την επεξεργασία που γίνεται στον πομπό. Δηλαδή η επεξεργασία στον δέκτη επαναφέρει την πληροφορία στην αρχική της κατάσταση, ώστε να μπορέσει να γίνει αντιληπτή από τον παραλήπτη. Όπως σημειώθηκε και στην εισαγωγή, η παρούσα εργασία αναφέρεται μόνο στον πομπό του συστήματος. Έτσι δεν θα δοθεί έμφαση στην λειτουργία του δέκτη και στην αντίστροφη επεξεργασία που ορίζεται από το πρωτόκολλο επικοινωνίας IEEE 802.11a.

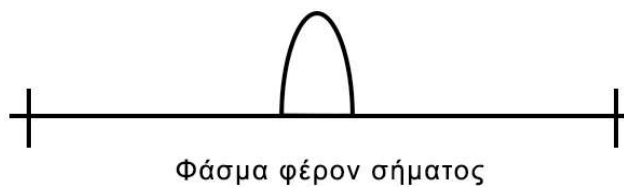
1.2 Ασύρματο τηλεπικοινωνιακό σύστημα

Βάσει του διαύλου επικοινωνίας τα τηλεπικοινωνιακά συστήματα χωρίζονται σε δύο κύριες κατηγορίες. Τα τηλεπικοινωνιακά συστήματα χωρίζονται σε ενσύρματα και σε ασύρματα συστήματα. Τα ενσύρματα συστήματα χρησιμοποιούν κάποιον δίαυλο επικοινωνίας που αποτελείται από ένα καλώδιο. Τα καλώδια που χρησιμοποιούνται πιο συχνά είναι τα καλώδια χαλκού και οι οπτικές ίνες. Από την άλλη πλευρά τα ασύρματα συστήματα δεν χρησιμοποιούν κάποιο μέσο σε στερεά μορφή. Το συνηθέστερο μέσο επικοινωνίας των ασύρματων συστημάτων είναι η ατμόσφαιρα. Και οι δύο κατηγορίες τηλεπικοινωνιακών συστημάτων ποικίλουν σε εφαρμογές και διαθέτουν μεγάλο πλήθος πρωτοκόλλων επικοινωνίας. Τα πρωτόκολλα επικοινωνίας περιγράφονται στη ενότητα 1.4.

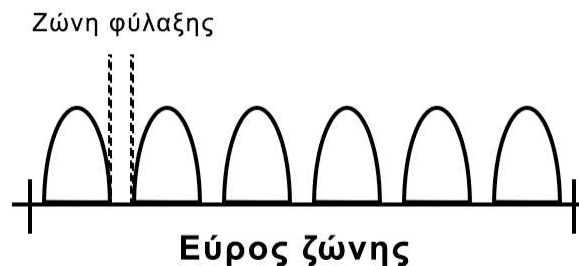
Ο πομπός που πρόκειται να υλοποιήσουμε αποτελεί τμήμα ενός ασύρματου τηλεπικοινωνιακού συστήματος. Το πρωτόκολλο IEEE 802.11a λαμβάνει υπόψιν του όλες τις παραμέτρους του δίαυλου επικοινωνίας και καθορίζει με ακρίβεια και πληρότητα το σύνολο των κανόνων που επιτρέπουν να ολοκληρωθεί επιτυχώς η ασύρματη επικοινωνία μεταξύ δύο σημείων. Επιπλέον, όπως θα δούμε στην επόμενη ενότητα (ενότητα 2) το πρωτόκολλο αυτό εκμεταλλεύεται τα χαρακτηριστικά του διαύλου επικοινωνίας ώστε να κάνει την επικοινωνία αποδοτικότερη.

1.3 Ορθογώνια πολυπλεξία με διαίρεση συχνότητας

Για την αποδοτικότερη χρήση του διαύλου επικοινωνίας χρησιμοποιούνται μέθοδοι πολυπλεξίας. Τα συνηθέστερα είδη πολυπλεξίας είναι η πολυπλεξία με διαίρεση χρόνου (time division multiplexing (TDM)) ή η πολυπλεξία με διαίρεση συχνότητας (frequency division multiplexing (FDM)). Για να μπορούν να αποσταλούν περισσότερα δεδομένα μέσα από τον ίδιο δίαυλο επικοινωνίας το πρωτόκολλο IEEE 802.11a χρησιμοποιεί FDM. Έτσι το υπόλοιπο της παραγράφου θα αφιερωθεί σε αυτήν. Η μέθοδος FDM χρησιμοποιεί ένα σύνολο φερόντων σημάτων, στα οποία ενσωματώνει την πληροφορία που πρέπει να μεταδώσει ο πομπός. Το εύρος ζώνης που ο πομπός έχει διαθέσιμο χωρίζεται σε ίσα τμήματα, τα οποία αποδίδονται στα φέρον σήματα. Το σύνολο των φερόντων σημάτων λέγεται σύμβολο. Στην πολυπλεξία αυτού του είδους τα φέρον σήματα του συμβόλου δεν αλληλεπικαλύπτονται. Έτσι όταν το φάσμα του φέρον σήματος είναι αυτό του σχήματος 2, τότε ο χωρισμός των διαθέσιμων συχνοτήτων και η δομή του συμβόλου στην FDM είναι αυτή του σχήματος 3.

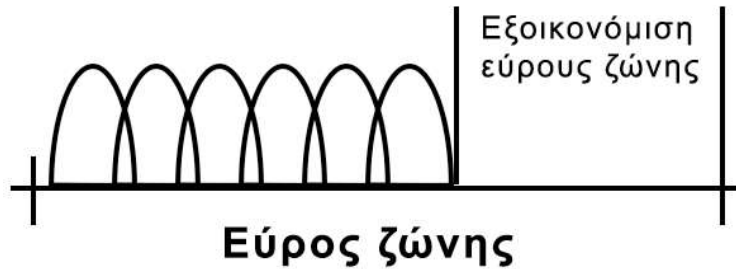


Σχήμα 2. Εύρος ζώνης ενός φέρον σήματος.



Σχήμα 3. Εύρος ζώνης όλων των φερόντων σημάτων στην πολυπλεξία με διαίρεση συχνότητας.

Η FDM δεν εκμεταλλεύεται πλήρως το σύνολο των συχνοτήτων που έχει στην διάθεσή της και ως αποτέλεσμα δεν αξιοποιείται αποδοτικά το φάσμα του συστήματος. Για αυτόν τον λόγο χρησιμοποιείται μία δεύτερη μέθοδος η οποία ονομάζεται ορθογώνια πολυπλεξία με διαίρεση συχνότητας (Orthogonal frequency division multiplexing (OFDM)), η οποία αποτελεί εξέλιξη της FDM. Στην OFDM τα φέρον σήματα του συμβόλου μπορούν και αλληλεπικαλύπτονται δίχως να παραβάλλονται. Αυτό συμβαίνει γιατί τα φέρον σήματα μιας OFDM διαθέτουν την ιδιότητα της ορθογωνικότητας, άρα είναι ανεξάρτητα μεταξύ τους. Η ιδιότητα αυτή επιτρέπει σε ένα σύνολο σημάτων να μεταδοθούν μέσω του ίδιου διαύλου επικοινωνίας και να ανιχνεύονται χωρίς να παρεμβάλλονται μεταξύ τους. Στο σχήμα 4 φαίνεται η επικάλυψη των σημάτων και η εξοικονόμηση μερικού εύρους ζώνης.



Σχήμα 4. Εύρος ζώνης όλων των φερόντων σημάτων στην ορθογώνια πολυπλεξία με διαίρεση συχνότητας.

Η OFDM έναντι στην FDM μπορεί και μεταδίδει την ίδια ποσότητα πληροφορίας χρησιμοποιώντας μικρότερο εύρος ζώνης. Αντίστοιχα, μπορεί και μεταδίδει περισσότερη πληροφορία για το ίδιο εύρος ζώνης.

1.4 Πρωτόκολλα επικοινωνίας

Στα τηλεπικοινωνιακά συστήματα ένα πρωτόκολλο είναι ένα σύνολο από κανόνες που καθορίζουν τον τρόπο με τον οποίο ανταλλάζουν δεδομένα οι σταθμοί ενός συστήματος επικοινωνίας. Για την επιτυχή επικοινωνία μεταξύ των σταθμών ενός συστήματος, οι σταθμοί αυτοί θα πρέπει να έχουν προσυμφωνήσει το πρωτόκολλο που θα χρησιμοποιήσουν. Το πρωτόκολλο καθορίζει ακριβώς τη μορφή, το χρόνο και τη σειρά που διακινούνται τα δεδομένα. Επίσης καθορίζει πως ανιχνεύεται και διορθώνεται ο θόρυβος που εμφανίζεται στα δεδομένα, καθώς και πώς αυτά μετατρέπονται ώστε να είναι κατάλληλα για διάδοση μέσω του διαύλου επικοινωνίας.

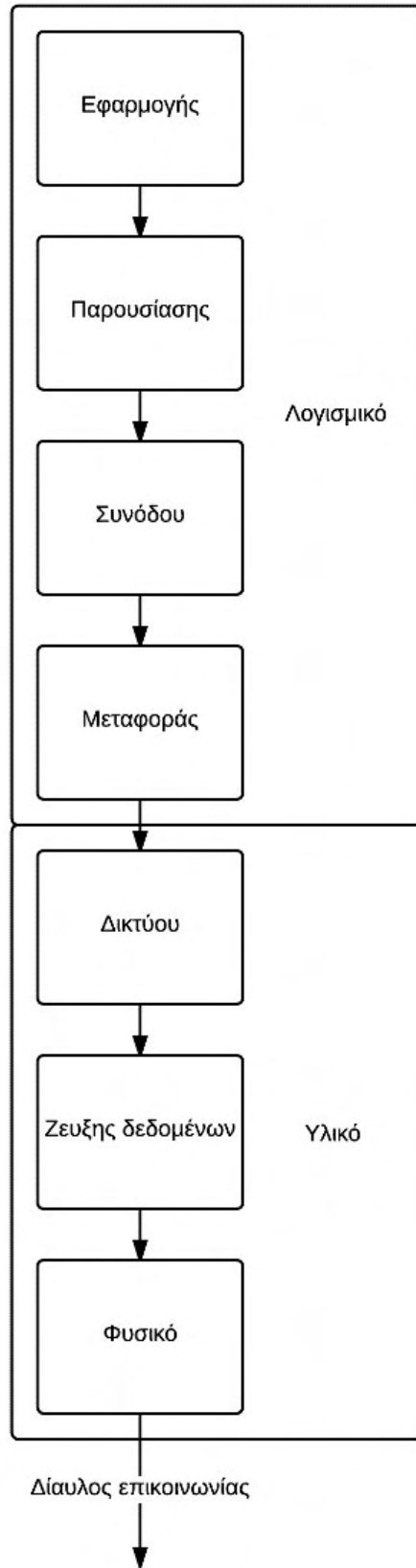
Εκτός από τα πρωτόκολλα που αναφέρονται στην μετάδοση πληροφορίας μεταξύ των σταθμών ενός συστήματος. Υπάρχουν και πρωτόκολλα που καθορίζουν τον τρόπο επικοινωνίας των εφαρμογών εντός των σταθμών αυτών. Τέτοιου είδους πρωτόκολλα λέγονται πρωτόκολλα υψηλού επιπέδου. Ενώ τα υπόλοιπα λέγονται χαμηλού επιπέδου. Ο διαχωρισμός των πρωτοκόλλων σε υψηλού και χαμηλού επιπέδου έγινε ώστε να χωριστεί το μεγάλο πρόβλημα σε μικρότερα και απλούστερα υποπροβλήματα. Αυτό διευκολύνει την προσθήκη και την βελτίωση των υπηρεσιών. Μία τέτοια δομή συνεργαζόμενων πρωτοκόλλων λέγεται οικογένεια πρωτοκόλλων και αποτελεί την αρχιτεκτονική ενός συστήματος επικοινωνίας.

1.3.1 Το μοντέλο OSI

Ο οργανισμός ISO (International Organization for Standardization) έχει ορίσει το πιο διαδεδομένο μοντέλο επικοινωνίας στα τηλεπικοινωνιακά συστήματα, το οποίο ονομάζεται OSI. Το μοντέλο OSI αποτελείται από 7 διαφορετικά στρώματα, τα οποία φαίνονται στο σχήμα 5. Τα πρώτα 4 επίπεδα καθορίζουν τα πρωτόκολλα επικοινωνίας που αφορούν τις εφαρμογές και τις συνδέσεις επικοινωνίας των σταθμών του συστήματος. Ενώ τα υπόλοιπα 3 επίπεδα αναφέρονται στον τρόπο αποστολής των δεδομένων μέσω του διαύλου επικοινωνίας.

Συγκεκριμένα, το επίπεδο εφαρμογής παρέχει στους χρήστες πρόσβαση στις υπηρεσίες δικτύου. Το επίπεδο παρουσίασης παρουσιάζει τα δεδομένα με κατάλληλο τρόπο και τα κωδικοποιεί ή τα αποκωδικοποιεί. Το επίπεδο συνόδου εκκινεί, διαχειρίζεται και τερματίζει την επικοινωνία. Το επίπεδο μεταφοράς χωρίζει το μήνυμα σε πακέτα και εξασφαλίζει ότι θα ληφθούν με την σωστή σειρά. Το επίπεδο δικτύου ορίζει τις διευθύνσεις αποστολής των πακέτων και καθορίζει την μεταφορά τους μέσω της κατάλληλης διαδρομής. Το επίπεδο ζεύξης δεδομένων καθορίζει μία αξιόπιστη σύνδεση μεταξύ δύο σταθμών. Ενώ το φυσικό επίπεδο διαχειρίζεται την φυσική μετάδοση των bit μέσω του διαύλου επικοινωνίας. Το κάθε ένα από αυτά είναι σχεδιασμένο ώστε να επικοινωνεί με τα δύο γειτονικά του επίπεδα. Έτσι στον πομπό τα δεδομένα ξεκινάνε από το πρώτο επίπεδο και να φτάνουν στο τελευταίο, στην συνέχεια μεταδίδονται μέσω του διαύλου επικοινωνίας και τελικά φτάνουν στον δέκτη, όπου από το τελευταίο επίπεδο φτάνουν στο πρώτο [2]. Η εργασία αυτή επικεντρώνεται στο φυσικό επίπεδο του μοντέλου αυτού και συγκεκριμένα στην πλευρά του πομπού.

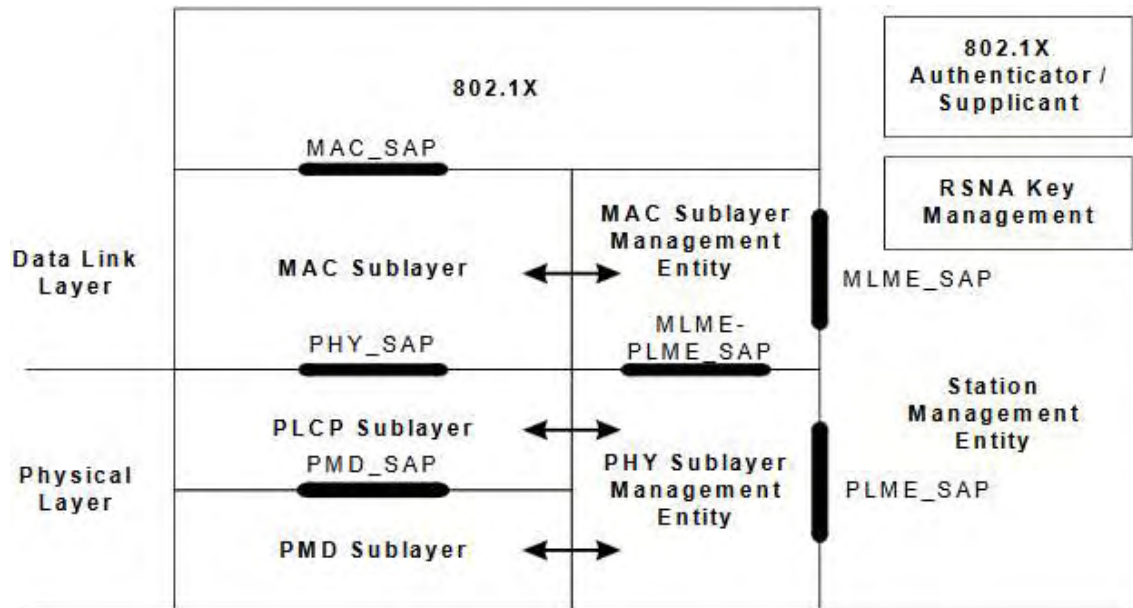
Για το φυσικό επίπεδο του μοντέλου OSI πολλές εταιρείες και οργανισμοί έχουν σχεδιάσει πρωτόκολλα επικοινωνίας. Ειδικά, η IEEE ορίζει πρωτόκολλα που αφορούν και ενσύρματες και ασύρματες επικοινωνίες. Για παράδειγμα το IEEE 802.3 είναι το συνηθέστερο πρωτόκολλο ενσύρματης δικτύωσης υπολογιστών, το οποίο είναι γνωστό ως Ethernet. Ενώ το IEEE 802.11 είναι μία οικογένεια πρωτοκόλλων για ασύρματα τοπικά δίκτυα (Local Area Networks (LAN)), που είναι γνωστή ως WiFi. Αυτή η οικογένεια παρέχει πρωτόκολλα που ποικίλουν στα χαρακτηριστήκα τους. Δηλαδή τα πρωτόκολλα της οικογένειας αυτή συνδυάζουν χαρακτηριστήκα όπως την μετάδοση της πληροφορίας σε μεγάλες αποστάσεις, την μετάδοση μεγάλης ποσότητας πληροφορίας σε μικρό χρονικό διάστημα, ή την μετάδοση με μικρό σχετικά κόστος. Μερικά από τα πρωτόκολλα του IEEE 802.11 είναι τα a, b, g, p, ac κ.λπ. Όπως έχει αναφερθεί πολλές φορές μέχρι τώρα, το φυσικό επίπεδο του πομπού που θα υλοποιηθεί ανήκει στο πρωτόκολλο IEEE 802.11a.



Σχήμα 5. Μοντέλο OSI.

2 Το πρωτόκολλο επικοινωνίας IEEE 802.11a

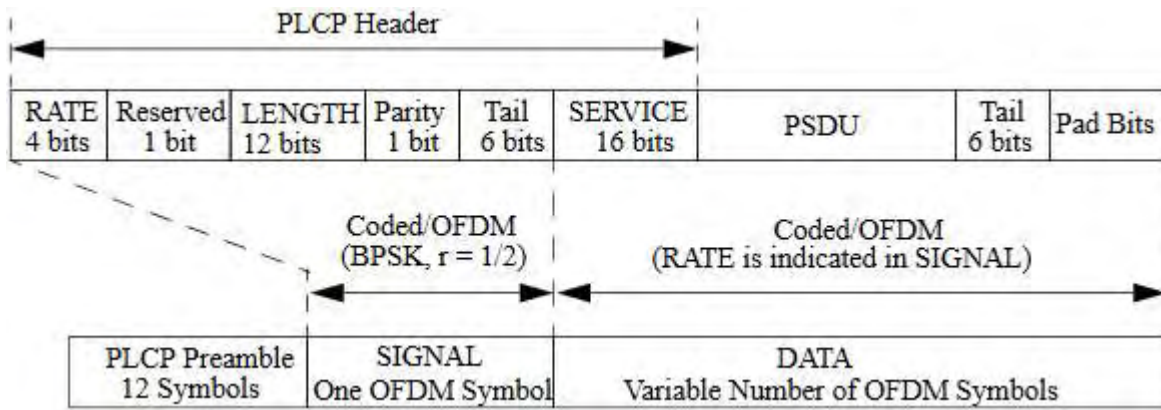
Το φυσικό επίπεδο τους συστήματος επικοινωνίας του πρωτοκόλλου IEEE 802.11a αποτελείται από τρία συνολικά υποεπίπεδα [1]. Τα δύο από αυτά εκτελούν λειτουργίες επεξεργασίας των δεδομένων, ώστε αυτά να είναι κατάλληλα για αποστολή και λήψη. Ενώ το τρίτο είναι το επίπεδο διαχείρισης, δηλαδή είναι αυτό που εξασφαλίζει την επικοινωνία μεταξύ των δύο άλλων επιπέδων. Η δομή του φυσικού επιπέδου φαίνεται στο σχήμα 6. (Το σχήμα αυτό προέρχεται από το αρχείο του πρωτοκόλλου IEEE 802.11a [1].)



Σχήμα 6. Δομή των επιπέδων Ζεύξης δεδομένων και Φυσικού για τις οικογένειες πρωτοκόλλων IEEE 802.1X.

Το πρώτο είναι το PLCP, το οποίο λαμβάνει το PSDU από το ανώτερο επίπεδο διαχείρισης πρόσβασης πολυμέσων (Medium Access Control (MAC)) και του προσαρτεί πληροφορίες που το καθιστούν κατάλληλο για αποστολή και λήψη. Το PSDU είναι το πακέτο που περιέχει την πληροφορία που θέλουμε να αποσταλεί. Το PSDU μαζί με τις προσαρτημένες πληροφορίες συνιστούν το PPDU, το οποίο είναι το συνολικό πακέτο δεδομένων πριν του προσαρτηθεί η κεφαλίδα και το πρόθεμα. Το τελικό πακέτο δεδομένων ονομάζεται πακέτο PLCP και περιγράφεται στην συνέχεια. Το PLCP ελαχιστοποιεί την εξάρτηση του MAC από το PMD (Physical Medium Dependent). Το PMD είναι το σύστημα που ορίζει τα χαρακτηριστικά και τις μεθόδους αποστολής και αποδοχής των πακέτων δεδομένων μεταξύ δύο ή περισσότερων σταθμών (STAs), μέσα από τον δίαυλο επικοινωνίας. Το PLME είναι το επίπεδο διαχείρισης των PLCP και PMD, αλλά ανταλλάζει και πληροφορίες με το layer διαχείρισης του MAC επιπέδου (MLME) τις οποίες τις χρησιμοποιεί έτσι ώστε το PLCP να κατασκευάσει το PPDU.

Στην παρούσα εργασία θα περιγραφεί, θα υλοποιηθεί και θα επαληθευτεί η σωστή λειτουργία του PLCP επιπέδου. Έτσι δεν θα δοθεί περαιτέρω επεξήγηση για τα υπόλοιπα επίπεδα του πρωτοκόλλου. Αυτό που πρέπει να υλοποιηθεί είναι η λήψη του PSDU πακέτου από το ανώτερο επίπεδο, η επεξεργασία του και η επέκτασή του με πληροφορία που το καθιστά ικανό για αποστολή. Το πακέτο PLCP στην τελική του μορφή, το οποίο είναι αυτό που προκύπτει στην έξοδο του συστήματος που υλοποιήθηκε, φαίνεται στο σχήμα 7. (Το σχήμα αυτό προέρχεται από το αρχείο του πρωτοκόλλου)



Σχήμα 7. Δομή του PLCP πακέτου.

Το πεδίο Preamble παρέχει την πληροφορία που καθιστά όλο το πακέτο ανιχνεύσιμο από τον δέκτη του συστήματος. Το πεδίο signal παρέχει τα μεταδεδομένα του πακέτου, δηλαδή παρέχει πληροφορίες όπως τον ρυθμό κώδικα βάση του οποίου πρέπει να αποκωδικοποιηθεί το υπόλοιπο πακέτο και το μήκος του πακέτου αυτού. Το πεδίο signal αποστέλλεται πάντα με διαμόρφωση BPSK και ρυθμό κώδικα $R = 1/2$, όπου είναι και ο πιο ανθεκτικός τρόπος αποστολής. Ενώ το πεδίο data περιέχει τα δεδομένα, το οποίο αποστέλλεται με τη διαμόρφωση και τον ρυθμό κώδικα που ορίζει το πεδίο rate του signal. Το κάθε ένα από τα πεδία που φαίνονται στο σχήμα 7 θα περιγράψουν στην επόμενη ενότητα (ενότητα 2.1).

Τέλος είναι σημαντικό να αναφερθεί ότι το πρωτόκολλο IEEE 802.11a χρησιμοποιεί OFDM για την μετάδοση των δεδομένων. Για την μετάδοση αυτή χρησιμοποιεί 64 φέρον σήματα, όπου στα 48 από αυτά κωδικοποιούνται δεδομένα, στα 4 κωδικοποιούνται πιλοτικές τιμές που βοηθούν στον συγχρονισμό του δέκτη με το PLCP πακέτο και στα υπόλοιπα αντιστοιχίζονται μηδενικές τιμές. Το πρωτόκολλο IEEE 802.11a έχει σχεδιαστεί με τρόπο που εκμεταλλεύεται πλήρως την OFDM μέθοδο και που λύνει μερικά προβλήματα που εμφανίζονται κατά την χρήση της.

2.1 Περιγραφή πεδίων του PLCP πακέτου

Το πρωτόκολλο IEEE 802.11a ορίζει ένα συγκεκριμένο πλήθος πεδίων που καθιστούν δυνατή την επικοινωνία. Τα τρία βασικά πεδία είναι το προοίμιο (preamble field), το πεδίου σήματος (signal field) και το πεδίο δεδομένων (data field). Το προοίμιο δεν περιέχει άλλα επιμέρους πεδία. Από την άλλη πλευρά τα πεδία signal και data χωρίζονται σε μικρότερα τμήματα, όπως ακριβώς φαίνεται και στην εικόνα 7. Το signal περιέχει τα πεδία rate, reserved, length, parity και tail. Ενώ το data περιέχει τα πεδία service, PSDU, tail και pad.

2.1.1 Πεδίο preamble

Ο δέκτης του συστήματος επικοινωνίας πρέπει να καταλάβει ότι το σήμα που λαμβάνει στην κεραία του, είναι σήμα πληροφορίας, πριν αρχίσει να το επεξεργάζεται. Για να γίνει δυνατή η ανίχνευση του σήματος πληροφορίας από τον δέκτη, ο πομπός του στέλνει πρώτα ένα δεύτερο σήμα. Το σήμα αυτό λέγεται πρόθεμα προοίμιο του πακέτου δεδομένων. Αν ο δέκτης λάβει και αναγνωρίσει το πρόθεμα του πακέτου δεδομένων, τότε θα καταλάβει ότι το σήμα που θα ακολουθήσει θα είναι το ίδιο το πακέτο. Στο πρωτόκολλο έχει οριστεί ακριβώς το μέγεθος και η μορφή του προθέματος που καθιστά δυνατή την σωστή αναγνώριση του πακέτου δεδομένων. Το πρόθεμα που έχει οριστεί από το πρωτόκολλο αποτελείται από δύο τμήματα. Το πρώτο στη σειρά λέγεται short OFDM training symbol, ενώ το δεύτερο λέγεται long OFDM training symbol. Για τον υπολογισμό των τμημάτων αυτών δίνονται οι σχέσεις 1 και 2 για το πρώτο και το δεύτερο τμήμα, αντίστοιχα.

Σχέση 1

$$S_{-26, 26} = \sqrt{(13/6)} * \{0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 0, 0, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0\}$$

Σχέση 2

$$L_{-26, 26} = \{1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 0, 1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1\}$$

Οι σχέσεις αυτές υποδεικνύουν τις τιμές των συμβόλων των δύο τμημάτων, οι οποίες ανήκουν στο πεδίο των συχνοτήτων.

Όπως φαίνεται και από την σχέση 1, στα περισσότερα υποφέρον σήματα του συμβόλου του τμήματος short OFDM training symbol θα αντιστοιχηθούν μηδενικές τιμές. Έτσι για να εξισορροπηθεί η μέση ενέργεια του συμβόλου, θα χρειαστεί οι υπόλοιπες μη μηδενικές τιμές να πολλαπλασιαστούν με τον παράγοντα κανονικοποίησης $\sqrt{(13/6)}$. Ενώ τα στοιχεία του long OFDM training symbol παραμένουν ως έχουν. Αυτή η δομή του προοιμίου δεν είναι η τελική. Για να εξαχθεί το προοίμιο όπως καθορίζει το πρωτόκολλο, πρέπει να περάσει από κάποια από τα συστήματα επεξεργασίας. Η επιπρόσθετη επεξεργασία του προοιμίου αλλά και η υλοποίηση του κυκλώματος που το εξάγει περιγράφεται στην ενότητα 2.3.9. Η τελική μορφή του προοιμίου φαίνεται στο σχήμα 18 της ίδιας ενότητας.

2.1.2 Πεδίο signal

Το πεδίο signal δεν περνάει από το κύκλωμα του scrambler, έτσι τα στοιχεία του δεν θα τυχαιοποιηθούν. Το σύστημα του scrambler περιγράφεται στην ενότητα 2.3.1. Η ροή επεξεργασίας του πεδίου signal αλλά και των υπόλοιπων πεδίων ελέγχεται από την μηχανή καταστάσεων του συστήματος, η οποία περιγράφεται στην ενότητα 4.2. Ενώ η ίδια η ροή δεδομένων περιγράφεται στην επόμενη ενότητα (ενότητα 2.2). Συνολικά το πεδίο signal απαρτίζεται από 24 bits μεταδεδωμένων. Τα πρώτα 4 bits, που απαριθμίζονται από το 0 έως το 3, συνιστούν το πεδίο rate. Το τέταρτο bit στην σειρά αποτελεί το πεδίο reserved, το οποίο θα έχει πάντα μηδενική τιμή και κρατιέται για μελλοντική χρήση. Τα 12 επόμενα bits, τα οποία απαριθμίζονται από 5 έως 16, απαρτίζουν το πεδίο length. Το πεδίο length καθορίζει το μήκος του πακέτου PSDU και μετριέται σε οκτάδες bit δεδομένων (octets). Στην συνέχεια βρίσκεται το 17ο bit το οποίο ονομάζεται πεδίο parity. Το πεδίο parity ορίζει αν το πλήθος των bit με μη μηδενική τιμή, των πεδίων rate και length, είναι ζυγός ή μονός αριθμός. Το parity bit χρησιμοποιείται για λόγους ανίχνευσης σφαλμάτων του πεδίου signal κατά την λήψη του από τον δέκτη του συστήματος. Τα τελευταία 6 bit ανήκουν στο πεδίο tail, το οποίο επιτρέπει την άμεση αποκωδικοποίηση των πεδίων rate και length, μετά την λήψη του signal. Αυτό χρησιμεύει γιατί η αποκωδικοποίηση του data προαπαιτεί τις πληροφορίες των πεδίων rate και length.

Όπως αναφέραμε και παραπάνω, το πεδίο signal κωδικοποιείται με ρυθμό κώδικα 1/2 και διαμορφώνεται με BPSK. Όμως το πεδίο data δεν έχει προκαθορισμένο ρυθμό κώδικα και τύπο διαμόρφωσης. Αυτά τα χαρακτηριστικά του πεδίου data ορίζονται από το πεδίο rate. Η διαμόρφωση και ο ρυθμός κώδικα που ορίζει η κάθε δυνατή τιμή του πεδίου rate φαίνεται στον πίνακα 1. Η συνολική δομή του πεδίου signal φαίνεται αναλυτικά στον πίνακα 2 και στο σχήμα 7 της ενότητας 2.

Πίνακας 1. Αντιστοίχιση των τιμών του πεδίου Signal με τους τύπους διαμόρφωσης και τους ρυθμούς κώδικα.

	Πεδίο rate	Διαμόρφωση	Ρυθμός κώδικα
1	1101	BPSK	1/2
2	1111	BPSK	3/4
3	0101	QPSK	1/2
4	0111	QPSK	3/4
5	1001	16-QAM	1/2
6	1011	16-QAM	3/4

7	0001	64-QAM	2/3
8	0011	64-QAM	3/4

Πίνακας 2. Πεδίο Signal.

Πεδίο rate 0 έως 3	Πεδίο reserved 4	Πεδίο length 5 έως 16	Πεδίο parity 17	Πεδίο tail 18 έως 23
-----------------------	---------------------	--------------------------	--------------------	-------------------------

2.1.3 Πεδίο data

Το πεδίο data έχει μεταβλητό μήκος. Το συνολικό του μήκος καθορίζεται από το μήκος του πακέτου PSDU και από τον τύπο διαμόρφωσης. Ο τύπος διαμόρφωσης επηρεάζει το μήκος του πεδίου data, γιατί το σύνολό του θα πρέπει να είναι πολλαπλάσιο της τιμής των bit δεδομένων ανά σύμβολο (N_{DBPS}). Η διαμόρφωση και το μέγεθος των bit δεδομένων ανά σύμβολο περιγράφονται στην ενότητα 2.3.5. Το πρώτο πεδίο του data είναι το πεδίο service, το οποίο απαρτίζεται από 16 μηδενικά bit. Τα πρώτα 6 bit του πεδίου service χρησιμοποιούνται για τον συγχρονισμό του descrambler στην πλευρά του συστήματος του δέκτη, ενώ τα υπόλοιπα 9 bit κρατιούνται για μελλοντική χρήση. Μετά το πεδίο PSDU τοποθετείται το πεδίο tail που έχει μέγεθος 6 bit. Το πεδίο tail έχει μηδενική τιμή και χρησιμοποιείται ώστε να επαναφερθεί ο συνελκτικός κωδικοποιητής στην αρχική του κατάσταση. Ενώ το πεδίο pad θα έχει τόσα bit, όσα χρειάζονται για να είναι το πεδίο data πολλαπλάσιο του αριθμού των bit δεδομένων που χωράνε σε ένα φέρον σήμα του συμβόλου. Όπως είπαμε και παραπάνω, το μήκος αυτό καθορίζεται από το μέγεθος του PSDU και από το είδος της διαμόρφωσης. Η συνολική δομή του πεδίου data φαίνεται στον πίνακα 3. Ενώ ο τρόπος που επηρεάζει το πεδίο rate τα μεγέθη του συστήματος φαίνονται στον πίνακα 4.

Πίνακας 3. Πεδίο Data.

Πεδίο service 0 έως 15 bit	Πεδίο PSDU 15 έως (length*8) + 15	Πεδίο tail (length*8) + 16 έως (length*8) + 21	Πεδίο pad (length*8) + 22 έως το τέλος
-------------------------------	--------------------------------------	--	--

Πίνακας 4. Αντιστοίχιση των τιμών του πεδίου Rate με τα μεγέθη N_{BPSC} , N_{CBPS} και N_{DBPS} .

	Πεδίο rate	Bits ανά υποφέρον σήμα (N_{BPSC})	Κωδικοποιημένα bits ανά σύμβολο (N_{CBPS})	Bits δεδομένων ανά σύμβολο (N_{DBPS})
1	1101	1	48	24
2	1111	1	48	36
3	0101	2	96	48
4	0111	2	96	72
5	1001	4	192	96
6	1011	4	192	144
7	0001	6	288	192
8	0011	6	288	216

Για την κατανόηση του τρόπου υπολογισμού και της χρησιμότητας του πεδίου pad δίνεται ένα παράδειγμα. Έστω ότι το μέγεθος του PSDU είναι ίσο με 800 bits. Ενώ Γνωρίζουμε ότι το πεδίο service έχει μέγεθος 16 bits και το πεδίο tail 6 bits. Επίσης σε αυτό το παράδειγμα, το πεδίο rate ισούται μια "1011" σε δυαδική βάση, άρα η διαμόρφωση ολόκληρου του πεδίου data θα είναι η 16-QAM, ο ρυθμός κώδικας θα είναι $r = 3/4$ και τα bit δεδομένων ανά σύμβολο θα είναι $N_{DBPS} = 144$. Τότε $(800 + 16 + 6) / N_{DBPS} = 5.708\dots$, άρα το

μέγεθος του πεδίου data δεν είναι πολλαπλάσιο των bit δεδομένων ανά σύμβολο. Για να διορθωθεί αυτό και για να μπορέσει να αποσταλθεί ολόκληρο το τελευταίο σύμβολο, θα πρέπει να υπολογιστούν τα αναγκαία pad bits. Το πλήθος των pad bit προέρχεται απευθείας από το MAC επίπεδο, επομένως δεν χρειάζεται να υπολογιστούν στο σύστημα του PLCP επιπέδου. Παρατηρείται ότι πρέπει να στρογγυλοποιηθεί προς τα πάνω το κλασματικό πηλίκο της προηγούμενης διαίρεσης. Έτσι το συνολικό μέγεθος του πεδίου data θα είναι ίσο με $N_{DBPS} * 6 = 864$. Άρα για τον υπολογισμό του πλήθους των pad bits πρέπει να αφαιρέσουμε από το συνολικό μέγεθος του πεδίου data, τα μεγέθη των πεδίων service, PSDU και tail. Έτσι το πεδίο pad θα έχει μέγεθος $864 - 822 = 42$ bits. Έτσι το πεδίο data θα μοιραστεί σε ακριβώς 6 σύμβολα δεδομένων. Επαναλήψεις training

2.2 Ροή δεδομένων του επιπέδου PLCP

Για την επεξεργασία και την αποστολή των δεδομένων, το πρωτόκολλο IEEE 802.11a έχει προκαθορίσει μία συγκεκριμένη ροή. Η ροή δεδομένων φαίνεται με ακρίβεια στα 13 βήματα που σημειώνονται στην συνέχεια:

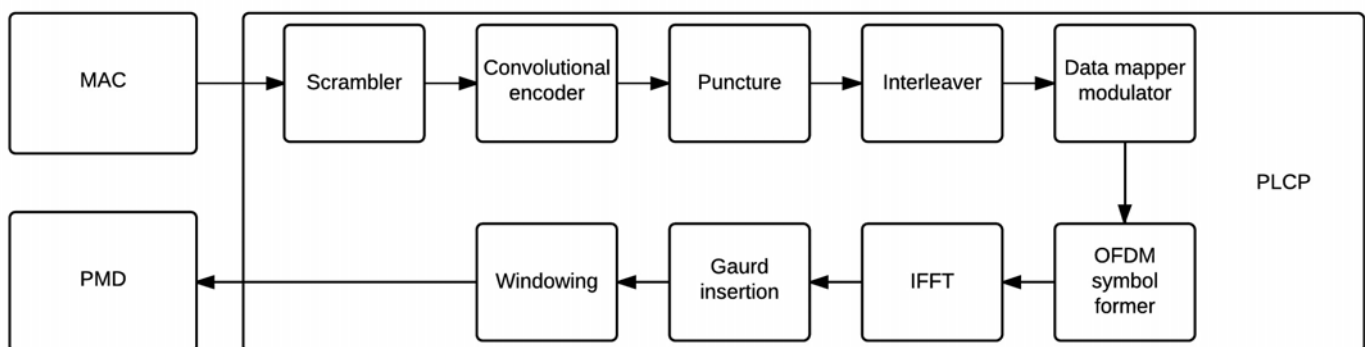
1. Οι τιμές συχνοτήτων του preamble είναι αποθηκευμένες σε μνήμες rom μέσα στο κύκλωμα του preamble. Στην συνέχεια οι τιμές αυτές περνάνε από το κύκλωμα του αντίστροφου γρήγορου μετασχηματισμού Fourier, ώστε να μεταφερθούν στο πεδίο του χρόνου. Πριν την αποστολή τους στον δέκτη, οι τιμές του προοιμίου στο πεδίο του χρόνου περνάνε και από το κύκλωμα εισαγωγής τιμών φύλαξης και παραθύρωσης. Το προοίμιο θα καταλήξει να αποτελείται από 10 επαναλήψεις μιας “short training sequence” και 2 επαναλήψεις μιας “long training sequence”, όπου οι δύο τελευταίες επαναλήψεις ακολουθούν το κυκλικό τους πρόθεμα. Πλέον, το preamble είναι έτοιμο για αποστολή στο επόμενο επίπεδο του συστήματος, δηλαδή στο PMD.
2. Το πεδίο signal συνενώνεται από το αντίστοιχο κύκλωμα, το οποίο απαρτίζεται από τα πεδία rate, reserved, PSDU_length, parity και tail.
3. Το πεδίο signal κωδικοποιείται με την βοήθεια του συνελκτικού κωδικοποιητή και περνάει αμετάβλητο από το σύστημα του puncture. Έτσι καταλήγει να έχει ρυθμό κώδικα ίσο με 1/2. Η διαμόρφωσή του θα είναι BPSK, οπότε θα περάσει και από τον interleaver με τον με block μεγέθους 48 bit και στην συνέχεια θα διαμορφωθεί. Μετά θα μεταφερθεί στο πεδίο του χρόνου με την βοήθεια του αντίστροφου γρήγορου μετασχηματισμού Fourier. Τέλος θα περάσει από το σύστημα εισαγωγής τιμών φύλαξης και παραθύρωσης, έτσι θα του προσαρτηθεί το κυκλικό πρόθεμα και θα υποδιπλασιαστούν κατάλληλα οι δύο του ακραίες τιμές. Πλέον, το πεδίο signal είναι έτοιμο για αποστολή στο επόμενο επίπεδο του συστήματος. Τονίζουμε ότι το πεδίο signal δεν περνάει από το κύκλωμα του scrambler.
4. Σύμφωνα με το πεδίο rate υπολογίζονται ο αριθμός των bit δεδομένων ανά OFDM σύμβολο (N_{DBPS}), ο ρυθμός κώδικα (r), ο αριθμός των bit ανά OFDM υποφέρον σήμα (N_{BPSK}) και ο αριθμός των κωδικοποιημένων bit ανά OFDM σύμβολο (N_{CBPS}).
5. Το πεδίο data συνενώνεται στο αντίστοιχο κύκλωμα. Συγκεκριμένα, το πεδίο PSDU προσαρτείται στο τέλος του πεδίου service, ακολουθούν τα πεδία tail και pad. Σημειώνουμε πως αν το τελικό μέγεθος του πεδίου data είναι πολλαπλάσιο του N_{DBPS} , τότε το πεδίο pad δεν θα χρειαστεί να προσαρτηθεί.
6. Πριν το πεδίο data αρχίσει να περνάει από το κύκλωμα του scrambler, αυτός αρχικοποιείται με έναν μη μηδενικό σπόρο. Συγκεκριμένα ο σπόρος που ορίζει το πρωτόκολλο είναι ο “1011101”. Στην συνέχεια το πεδίο data περνάει από αυτόν και τυχαιοποιούνται τα δεδομένα του.
7. Μετά την χρήση του scrambler τα έξι bit του πεδίου tail μηδενίζονται, ώστε να μπορέσει να μηδενιστεί και ο συνελκτικός κωδικοποιητής.
8. Το τυχαιοποιημένο σήμα δεδομένων περνάει από τον συνελκτικό κωδικοποιητή και μετά από το κύκλωμα του puncture. Έτσι καταλήγει να έχει κωδικοποιηθεί και να έχει μεταβληθεί ο ρυθμός κώδικα του όπως ορίζει το πεδίο rate.

9. Το κωδικοποιημένο σήμα χωρίζεται σε ομάδες των N_{CBPS} bits, το οποίο ορίζεται και αυτό από το πεδίο rate. Σε κάθε μία ομάδα από αυτές γίνεται interleaving, δηλαδή τα δεδομένα της κάθε ομάδας αλλάζουν σειρά με συγκεκριμένο τρόπο.
10. Το κωδικοποιημένο και interleaved σήμα, από το προηγούμενο βήμα, ξαναχωρίζεται σε ομάδες των N_{BPS} bits. Η κάθε μία ομάδα περνάει από το κύκλωμα διαμόρφωσης. Έτσι εξάγονται μεμονωμένοι διαμορφωμένοι αριθμοί, οι οποίοι είναι μιγαδικοί.
11. Οι διαμορφωμένες τιμές περνάνε από το κύκλωμα symbol former, στο οποίο το σύνολο τους χωρίζεται σε ομάδες των 48. Ο κάθε ένας αριθμός των ομάδων απαριθμίζεται από το 0 έως το 47 και στην συνέχεια δίνεται στα υποφέρον σήματα -26 έως -22, -20 έως -8, -6 έως -1, 1 έως 6, 8 έως 20 και 22 έως 26 του συμβόλου, αντίστοιχα.
12. Στα 4 υποφέρον σήματα -21, -7, 7 και 21 εισάγονται πιλοτικές τιμές. Ο συνολικός αριθμός των υποφέρον σημάτων είναι 52 (48 τιμές δεδομένων + 4 πιλοτικές τιμές). Το σύνολο των 52 υποφέρον σημάτων απαρτίζουν ένα σύμβολο.
13. Κάθε σύμβολο μεταφέρεται στο πεδίο του χρόνου μέσω του αντίστροφου γρήγορου μετασχηματισμού Fourier. Στην έξοδο του κυκλώματος του μετασχηματισμού δίνεται ένα επεκταμένο σύμβολο 64 υποφέρον σημάτων, των οποίων οι τιμές ανήκουν στο πεδίο του χρόνου. Στην συνέχεια, το κάθε επεκταμένο σύμβολο περνάει από το κύκλωμα εισαγωγής τιμών φύλαξης και παραθύρωσης. Έτσι στο καθένα από αυτά τοποθετείται μία κυκλική προέκταση, η οποία ονομάζεται πρόθεμα τιμών φύλαξης. Ενώ λόγω της παραθύρωσης περικύπτει το προκύπτων περιοδικό κύμα και σχηματίζεται μόνο ένα OFDM σύμβολο. Πλέον και το πεδίο data είναι έτοιμο για αποστολή στο επόμενο επίπεδο του συστήματος επικοινωνίας.

Η ροή αυτή επιτυγχάνεται με την βοήθεια μιας μηχανής καταστάσεων. Η υλοποιημένη μηχανή καταστάσεων περιγράφεται στην ενότητα 4.2.

2.3 Τα συστήματα του επιπέδου PLCP

Το πρωτόκολλο IEEE 802.11a έχει προκαθορισμένη μία σειρά από συστήματα που επεξεργάζονται τα δεδομένα με κατάλληλο τρόπο. Όλα τα συστήματα που ορίζει το πρωτόκολλο περιγράφονται σε αυτή την ενότητα. Τα συστήματα αυτά είναι ο scrambler, ο συνελκτικός κωδικοποιητής, ο puncture, ο interleaver, ο διαμορφωτής, ο symbol former, το σύστημα πιλοτικών τιμών, ο αντίστροφος γρήγορος μετασχηματισμός Fourier, η εισαγωγή κυκλικού προθέματος και η παραθύρωση. Η βασική δομή του PLCP επιπέδου φαίνεται στο διάγραμμα του σχήματος 8.



Σχήμα 8. Δομή και ροή πληροφορίας στο PLCP επίπεδο.

2.3.1 Scrambler

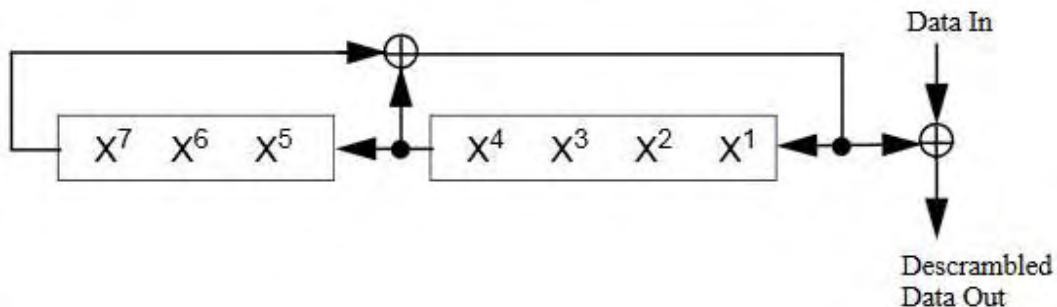
Η γενική λειτουργία του scrambler είναι ο μετασχηματισμός ενός σήματος πληροφορίας, σύμφωνα με ένα δεύτερο προκαθορισμένο σήμα που παράγεται εσωτερικά του συστήματος. Τα δεδομένα που περιέχει το σήμα πληροφορίας θα αποσταλθούν στον δέκτη του συστήματος. Το ίδιο το σήμα πληροφορίας αποτελεί την είσοδο του scrambler. Αντίστοιχα η έξοδος του, απαρτίζεται από το τελικό μετασχηματισμένο σήμα. Το μέγεθος του σήματος εξόδου είναι ίσο με το μέγεθος του σήματος πληροφορίας. Το εσωτερικό σήμα λέγεται σήμα μετασχηματισμού ή φέρον σήμα (carrier), μιας και βάσει αυτού το σήμα πληροφορίας μετατρέπεται στο επιθυμητό σήμα εξόδου. Η λειτουργία του scrambler εξαρτάται πλήρως από το σήμα μετασχηματισμού. Οπότε με την παραγωγή κατάλληλου σήματος μετασχηματισμού μπορεί να παραχθεί και η επιθυμητή τελική μορφή του σήματος πληροφορίας. Οι συνηθέστεροι μετασχηματισμοί που κάνει ένας scrambler στο σήμα πληροφορίας είναι η αντιστροφή (inverse), η κωδικοποίηση (encode) ή η τυχαιοποίηση (randomize) [3]. Κατά την διαδικασία της αντιστροφής όλα τα μηδενικά bit του σήματος μετατρέπονται σε άσους και όλοι οι άσοι μετατρέπονται σε μηδενικά. Η κωδικοποίηση παραμορφώνει το σήμα πληροφορίας με τρόπο τέτοιο ώστε να μην γίνονται αντιληπτά τα δεδομένα του. Η κωδικοποίηση εφαρμόζεται σε συστήματα κρυπτογράφησης. Η τυχαιοποίηση χρησιμοποιεί ένα ψευδοτυχαίο σήμα μετασχηματισμού και μετατρέπει το σήμα πληροφορίας σε ένα δεύτερο ψευδοτυχαίο σήμα. Ο scrambler αποτελεί τμήμα του πομπού ενός συστήματος επικοινωνίας. Αντίστοιχα, ο δέκτης περιλαμβάνει το σύστημα που επαναφέρει το σήμα πληροφορίας στην αρχική του κατάσταση, το οποίο ονομάζεται descrambler.

Το πρωτόκολλο IEEE 802.11a χρησιμοποιεί ένα σύγχρονο scrambler πλαισίου. Αυτό σημαίνει ότι το σήμα μετασχηματισμού που παράγει επαναλαμβάνονται συνεχώς, όπως θα δούμε και στην συνέχεια. Προκειμένου να λυθούν συγκεκριμένα προβλήματα της επικοινωνίας, ο scrambler λειτουργεί ως τυχαιοποιητής. Τα προβλήματα που πρόκειται να επιλυθούν με την βοήθεια του scrambler περιγράφονται στην επόμενη παράγραφο. Όπως αναφέραμε προηγουμένως, η λειτουργία του scrambler εξαρτάται πλήρως από το σήμα μετασχηματισμού. Για να επιτευχθεί η σωστή λειτουργία του scrambler ως τυχαιοποιητής, πρέπει αυτός να παράγει ένα ψευδοτυχαίο σήμα μετασχηματισμού. Για τον σκοπό αυτό το πρωτόκολλο χρησιμοποιεί μια πολυωνυμική γεννήτρια (generator polynomial). Το πολυώνυμο που χρησιμοποιείται για την παραγωγή του ψευδοτυχαίου σήματος μετασχηματισμού δίνεται από την σχέση 3.

Σχέση 3

$$S(x) = x^7 + x^4 + 1$$

Το πολυώνυμο αυτό υλοποιείται με ένα καταχωρητή ολίσθησης (shift register) και μία λογική πύλη XOR, όπως φαίνεται και στο σχήμα 9.



Σχήμα 9. Αρχιτεκτονική του Scrambler.

Για την παράγωγή του σήματος μετασχηματισμού, ο καταχωρητής ολίσθησης τροφοδοτείται με έναν μη μηδενικό σπόρο. Αν ο σπόρος είναι μηδενικός, τότε το σήμα εξόδου θα είναι αυτούσιο το σήμα εισόδου, γιατί η

λογική πράξη XOR ανάμεσα στο μηδέν και σε ένα άλλο σήμα θα δώσει ως αποτέλεσμα την τιμή του δεύτερου σήματος. Κάτι τέτοιο δεν είναι επιθυμητό, για αυτόν τον λόγο το πρωτόκολλο ορίζει τον σπόρο ως μη μηδενικό. Η ψευδοτυχαία ακολουθία που θα παραχθεί θα έχει μέγεθος 127 bit, αυτό σημαίνει ότι θα παραχθούν 127 ψευδοτυχαία bit πριν αρχίσει το σήμα από την αρχή. Δηλαδή όπως είπαμε, το σήμα επαναλαμβάνεται περιοδικά κάθε 127 bit. Αφού παραχθεί με επιτυχία το ζητούμενο σήμα μετασχηματισμού, πρέπει να σχηματίσει και το τυχαίο σήμα εξόδου. Για να γίνει αυτό, κάθε χρονική στιγμή το σήμα μετασχηματισμού και το σήμα εισόδου περνάνε από μία λογική πύλη XOR, το αποτέλεσμα της οποίας αποστέλλεται στην έξοδο του scrambler. Το σχηματικό διάγραμμα, που δίνεται από το πρωτόκολλο, του scrambler φαίνεται στο σχήμα 9. (Το σχήμα αυτό προέρχεται από το αρχείο του πρωτοκόλλου [1].) Ο scrambler θα λαμβάνει απευθείας ως είσοδο τα bit που προέρχονται από την συνένωση των πεδίων του Service, PSDU, tail και των pad bit. Στην συνέχεια η έξοδος του θα αποστέλλεται στην είσοδο του συνελκτικού κωδικοποιητή (convolutional encoder), ο οποίος περιγράφεται στην ενότητα 2.3.2. Είναι σημαντικό να πούμε ότι πρέπει να μηδενιστεί το τμήμα του πεδίου tail μετά την τυχαιοποίησή του, έτσι ώστε να μπορεί να επανεκκινείται ο συνελκτικός κωδικοποιητής που ακολουθεί [1].

Αν το σήμα δεδομένων αποστέλλεται στην αρχική του κατάσταση, δηλαδή δίχως να μετασχηματιστεί με την βοήθεια του scrambler, τότε θα ήταν πολύ πιθανό να περιέχει μεγάλες ακολουθίες συνεχόμενων μηδενικών ή μοναδιαίων bit (whitening sequences). Αυτό θα σήμαινε ότι η ενέργειά του θα ήταν συσσωρευμένη σε συγκεκριμένες περιοχές και ότι θα έχει χαμηλή εντροπία. Άρα δεν θα ικανοποιούσε και τις απαιτήσεις της μέγιστης πυκνότητας φασματικής ισχύος. Έτσι δεν θα ήταν κατάλληλο για αποστολή, γιατί όταν η ενέργεια είναι συσσωρευμένη σε μία στενή ζώνη συχνοτήτων υπάρχει μεγάλη πιθανότητα εμφάνισης του φαινομένου διασταυρούμενης διαμόρφωσης (cross modulation). Δηλαδή το σήμα προς αποστολή πιθανότατα δεν θα λαμβανόταν όπως στάλθηκε, λόγω του θορύβου που θα προστιγόταν στα γειτονικά υποφέρον σήματα από αυτά με την συσσωρευμένη ενέργεια. Το φαινόμενο αυτό ονομάζεται διαφερωντική παρεμβολή (inter-subcarrier interference (ICI)) [4]. Όμως με την χρήση του scrambler ως τυχαιοποιητή, η διασπορά της ενέργειας στις συχνότητες του σήματος πληροφορίας είναι μεγαλύτερη και έτσι το σήμα μεταβαίνει σε κατάσταση κατάλληλη για μετάδοση. Για την επιπρόσθετη καταπολέμηση της διαφερωντικής παρεμβολής χρησιμοποιείται ένα ακόμα κύκλωμα, το οποίο λέγεται σύστημα προσθήκης κυκλικού προθέματος και περιγράφεται στην ενότητα 2.3.9. Οπότε η τυχαιοποίηση του σήματος δεδομένων κρίνεται απαραίτητη, προκειμένου να πάρει μορφή που είναι κατάλληλη για μετάδοση, δηλαδή να έχει υψηλή εντροπία.

2.3.2 Συνελκτικός κωδικοποιητής

Ένας συνελκτικός κωδικοποιητής (convolutional encoder) είναι ένα σύστημα ανίχνευσης και διόρθωσης λαθών για κανάλια χωρίς μνήμη. Το σήμα πληροφορίας που λαμβάνει στην είσοδό του κωδικοποιείται ώστε να επιτευχθεί η ανίχνευση και η διόρθωση λαθών που μπορεί να προκύψουν κατά την αποστολή του. Για την μεγαλύτερη ανθεκτικότητα του σήματος προς αποστολή, οι συνελκτικοί κωδικοποιητές παράγουν περισσότερη πληροφορία. Αυτό σημαίνει ότι για κάθε bit δεδομένων εισόδου, ο κωδικοποιητής παράγει περισσότερα από ένα bit, κάθε χρονική στιγμή. Για παράδειγμα, ένας συνελκτικός κωδικοποιητής με ρυθμό (rate) 1/2, κάθε χρονική στιγμή λαμβάνει στην είσοδό του ένα bit πληροφορίας και κωδικοποιώντας την παράγει δύο bit κωδικοποιημένης πληροφορίας. Ο ρυθμός μετριέται σε bits ανά σύμβολο. Οπότε στο παράδειγμα αυτό, όπου έχουμε έναν συνελκτικό κωδικοποιητή ρυθμού κώδικα ίσο με 1/2, σε ένα σύμβολο θα χωράνε τόσα bit δεδομένων όσο είναι το μισό του μέγεθος. Δηλαδή, σε ένα σύμβολο με 48 υποφέρον σήματα θα χωράνε 24 bit πληροφορίας. Γενικά, ένας συνελκτικός κωδικοποιητής μπορεί να εξάγει όσα bit προσδιορίζει η αρχιτεκτονική του. Έτσι, μπορούν και σχεδιάζονται και άλλοι κωδικοποιητές με μεγαλύτερο ρυθμό, αναλόγως με τις απαιτήσεις του κάθε συστήματος. Εκτός όμως από τον ρυθμό κώδικα, υπάρχει και ένα ακόμα μέγεθος που χαρακτηρίζει έναν συνελκτικό κωδικοποιητή, αυτό είναι το μέγεθος της μνήμης του. Η μνήμη αποθηκεύει ένα πλήθος από προηγούμενες εισόδους του συστήματος. Σε ένα σύστημα με μνήμη μεγέθους k , η έξοδος του κάθε χρονική στιγμή εξαρτάται από την τρέχον είσοδο αλλά και από τις τελευταίες προηγούμενες k εισόδους. Στις περισσότερες σχεδιάσεις συνελκτικών κωδικοποιητών η μετατροπή του μηνύματος εξαρτάται πλήρως και από προηγούμενες εισόδους του συστήματος.

Ο συνελκτικός κωδικοποιητής που χρησιμοποιείται από το πρωτόκολλο έχει ρυθμό κώδικα 1/2. Για την αύξηση του ρυθμού αυτού, προκειμένου να χωράνε περισσότερα bit δεδομένων μέσα σε ένα σύμβολο, θα προστεθεί ένα ακόμα κύκλωμα αμέσως μετά τον συνελκτικό κωδικοποιητή. Το κύκλωμα αυτό ονομάζεται puncture και η λειτουργία του θα περιγραφεί αναλυτικά στην επόμενη ενότητα (ενότητα 2.3.3). Η χρήση του puncture θα επηρεάσει ανάλογα την ανθεκτικότητα του συστήματος. Δηλαδή, όσα λιγότερα είναι τα κωδικοποιημένα bit ενός συμβόλου, τόσο πιο ευάλωτο είναι σε σφάλματα. Η μνήμη του συνελκτικού κωδικοποιητή του πρωτοκόλλου έχει μήκος 6 στοιχείων, έτσι κάθε χρονική στιγμή η έξοδος του εξαρτάται από την τρέχον αλλά και από τις προηγούμενες 6 εισόδους. Η σχεδίαση του συνελκτικού κωδικοποιητή είναι απλή και το σχηματικό του διάγραμμα φαίνεται στο σχήμα 10. Παρατηρείται πως για την παραγωγή των σημάτων εξόδου, χρησιμοποιείται η ίδια τεχνική που χρησιμοποιείται και στον scrambler για την παραγωγή του σήματος μετασχηματισμού του scrambler. Δηλαδή, για την παραγωγή της τιμής των σημάτων εξόδου χρησιμοποιεί ένα γεννήτρια πολυώνυμο. Η πρώτη έξοδος, που στο σχήμα ονομάζεται “Α”, χρησιμοποιεί το πολυώνυμο της σχέσης 4 για την παραγωγή της τιμής του. Αντίστοιχα, η δεύτερη έξοδος, με το όνομα “Β” στο σχήμα, χρησιμοποιεί το πολυώνυμο της σχέσης 5.

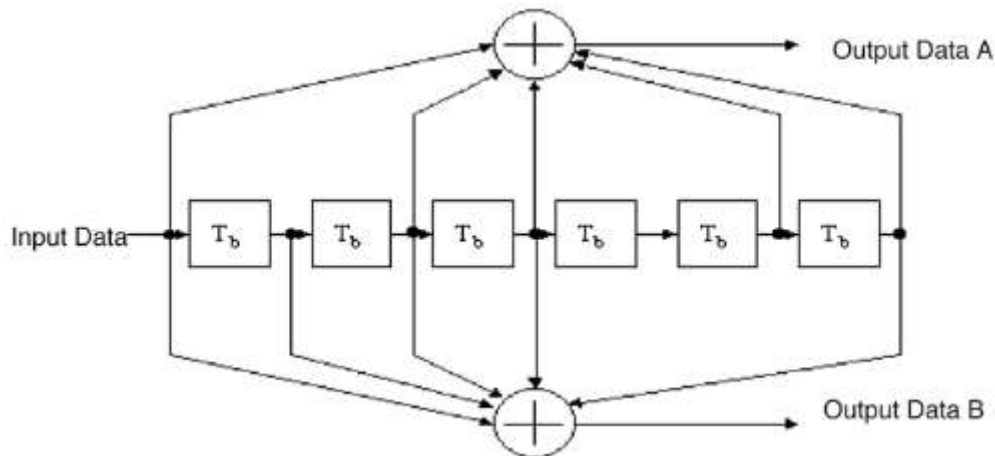
Σχέση 4

$$S_A(x) = x^6 + x^5 + x^3 + x^2 + 1$$

Σχέση 5

$$S_B(x) = x^6 + x^3 + x^2 + x + 1$$

Πρέπει να τονιστεί πως είτε σε σειριακή είτε σε παράλληλη έξοδο, “το σήμα Α θα βρίσκεται πρώτο στην σειρά”, όπως αναφέρεται και στο πρωτόκολλο [1]. Αναφέρουμε ότι ο αποκωδικοποιητής που προτείνει το πρωτόκολλο είναι ο αλγόριθμος Viterbi [9].



Σχήμα 10. Αρχιτεκτονική του συνελκτικού κωδικοποιητή.

Για την σωστή λειτουργία και την παραγωγή των προσδοκώμενων αποτελεσμάτων, η μνήμη του συνελκτικού κωδικοποιητή πρέπει να επανεκκινηθεί προτού αρχίσει να κωδικοποιεί τα δεδομένα. Όμως, όπως είπαμε και κατά την περιγραφή του scrambler, το πεδίο tail του PPDU χρησιμοποιείται ακριβώς για αυτόν τον σκοπό. Δηλαδή, εφόσον θα είναι μηδενικό και αφού έχει μήκος 6 bit, θα εισχωρήσει σε όλον τον καταχωρητή ολίσθησης του συνελκτικού κωδικοποιητή. Έτσι, για την χρονική στιγμή ακριβώς πριν την εισαγωγή του μηνύματος εισόδου, η μνήμη του κωδικοποιητή θα είναι μηδενισμένη. Επομένως, όταν θα λάβει το μήνυμα εισόδου θα παραγάγει την κωδικοποιημένη πληροφορία που αναμένεται. Ο μηδενισμός του πεδίο tail του Data αποτρέπει την επιπρόσθετη ανάπτυξη του συστήματος του συνελκτικού κωδικοποιητή. Βάσει πρωτοκόλλου, ο

συνελικτικός κωδικοποιητής τοποθετείται ακριβώς μετά τον scrambler και στέλνει την έξοδο του απευθείας στην είσοδο του puncture.

Ο συνελικτικός κωδικοποιητής δεν έχει ως στόχο την κρυπτογράφηση του μηνύματος που λαμβάνει, άλλα προορίζεται ξεκάθαρα για την ανίχνευση και την διόρθωση λαθών που μπορεί να προκύψουν κατά την αποστολή του μηνύματος αυτού. Η χρήση του εξασφαλίζει ότι το τελικό μήνυμα θα είναι ανθεκτικό και ότι θα αποσταλθεί με αξιοπιστία. Πιο συγκεκριμένα, ο συνελικτικός κωδικοποιητής προορίζεται για ανίχνευση και διόρθωση τυχαίων σφαλμάτων (random errors), τα οποία προκαλούνται σε κανάλια χωρίς μνήμη. Αν στο σήμα πληροφορίας εισαχθεί θόρυβος κατά την αποστολή, ο οποίος θα επηρεάσει ένα μεγαλύτερο σύνολο bit, από αυτό που μπορεί να διορθώσει, τότε ο αποκωδικοποιητής του δέκτη δεν θα είναι ικανός να διορθώσει αυτού του είδους το σφάλμα [4]. Το σφάλμα ενός συνόλου συνεχόμενων bit ονομάζεται σφάλμα έκρηξης (burst error). Τα σφάλματα έκρηξης αντιμετωπίζονται διαφορετικά από ένα άλλο σύστημα, το οποίο λέγεται interleaver και θα το περιγράψουμε στην ενότητα 2.3.4. Ο συνελικτικός κωδικοποιητής αντεπεξέρχεται στις απαιτήσεις του συστήματος και αν προκύψει ένα τυχαίο σφάλμα τότε, σε συνδυασμό με τον αποκωδικοποιητή του δέκτη, θα ανιχνεύσει και θα διορθώσει το σφάλμα με επιτυχία. Το πρωτόκολλο επιλέγει αυτόν το τύπο κωδικοποιητή γιατί απαιτεί την σειριακή κωδικοποίηση των δεδομένων. Αν επέλεγε ένα τμηματικό κωδικοποιητή θα αναγκαζόταν να συλλέγει και να αποθηκεύει ένα τμήμα του μηνύματος πριν το κωδικοποιήσει, πράγμα που θα αύξανε την πολυπλοκότητα και θα μείωνε την απόδοση του συστήματος του πομπού [4]. Σε κάθε περίπτωση, αν το σύστημα του πομπού δεν ενσωμάτωνε κάποιον κωδικοποιητή, η επικοινωνία θα ήταν αδύνατη. Αυτό θα συνέβαινε γιατί και το πιο μικρό σε μέγεθος τυχαίο σφάλμα θα την εμπόδιζε. Επομένως, η χρήση ενός συστήματος ανίχνευσης και διόρθωσης τυχαίων σφαλμάτων κρίνεται αναγκαία προκειμένου να επιτευχθεί η επικοινωνία.

2.3.3 Puncture

Η λειτουργία του puncture είναι η κατ' επιλογήν μεταβολή του ρυθμού κώδικα του συστήματος του πομπού. Η είσοδός του προέρχεται απευθείας από κάποιον κωδικοποιητή. Έτσι, θα έχει ρυθμό δεδομένων εισόδου ίσο με αυτόν του κωδικοποιητή του συστήματος. Για την χρήση ενός συγκεκριμένου ρυθμού κώδικα, ένα σύστημα δεν θα χρειαζόταν να ενσωματώνει το puncture, απλώς θα έπρεπε να σχεδιάζε τον συνελικτικό κωδικοποιητή με τέτοιο τρόπο που θα κωδικοποιούσε το σήμα πληροφορίας με τον ζητούμενο ρυθμό κώδικα. Όμως η χρήση του puncture αυξάνει την ευελιξία του συστήματος ως προς τον χειρισμό αυτού του ρυθμού. Για την ακρίβεια, ο puncture μπορεί είτε να αυξήσει είτε να συντηρήσει τον ρυθμό κώδικα του συνελικτικού κωδικοποιητή. Και κατά συνέπεια, μπορεί είτε να αυξηθεί, είτε να συντηρηθεί και ο ρυθμός μετάδοσης δεδομένων. Πρέπει να πούμε ότι το βασικό σύστημα που χειρίζεται τον ρυθμό μετάδοσης δεδομένων είναι το σύστημα διαμόρφωσης, το οποίο θα περιγραφεί στην ενότητα 2.3.5. Όμως όπως είπαμε, η χρήση του puncture αυξάνει την ευελιξία του πομπού, μιας και είναι το μόνο σύστημα που μεταβάλλει τον ρυθμό κώδικα.

Για καλύτερη κατανόηση της χρήσης του puncture δίνεται ένα παράδειγμα. Σε ορισμένες καταστάσεις περιβάλλοντος με υψηλή πιθανότητα εμφάνισης σφαλμάτων στο σήμα πληροφορίας, ο puncture μπορεί να διατηρήσει τον ρυθμό κώδικα του συνελικτικού κωδικοποιητή, ώστε το σήμα πληροφορίας που πρόκειται να αποσταλθεί να είναι όσο πιο ανθεκτικό μπορεί να γίνει. Από την άλλη πλευρά, σε περιπτώσεις επικοινωνίας που απαιτείται γρήγορη μετάδοση της πληροφορίας και στο περιβάλλον το οποίο είναι εγκατεστημένο το σύστημα δεν υπάρχει μεγάλη πιθανότητα εμφάνισης σφαλμάτων, ο puncture μπορεί να αυξήσει τον ρυθμό κώδικα ούτως ώστε να μπορέσουν να αποσταλούν περισσότερα δεδομένα ανά σύμβολο.

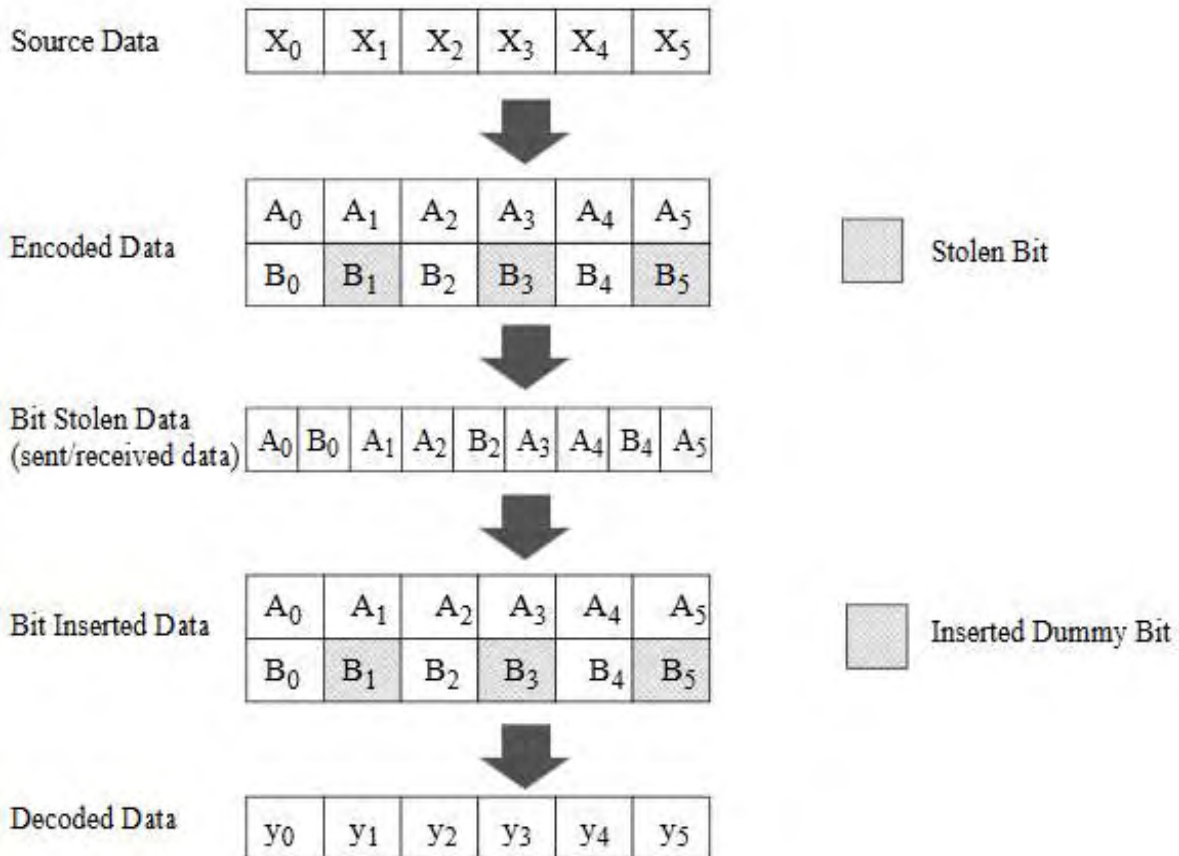
Το πρωτόκολλο 802.11a παρουσιάζει ένα σύστημα puncture τριών καταστάσεων. Η πρώτη κατάσταση εξάγει το κωδικοποιημένο σήμα αυτούσιο με ρυθμό κώδικα 1/2. Η δεύτερη αφαιρεί κάποια bits από την ακολουθία του κωδικοποιημένου σήματος και μεταβάλλει την τιμή του ρυθμού κώδικα σε 2/3 bit ανά σύμβολο. Ενώ στην τρίτη και τελευταία κατάσταση μειώνει λίγο ακόμα το πλήθος των κωδικοποιημένων bit, έτσι αυξάνεται και η τιμή του ρυθμού κώδικα, που φτάνει τα 3/4 bits ανά σύμβολο. Οι ρυθμοί κώδικα που υποστηρίζει το πρωτόκολλο φαίνονται στον πίνακα 5.

Πίνακας 5. Οι δυνατοί ρυθμοί κώδικα που μπορεί να παραχθούν από το σύστημα του Puncture.

	Ρυθμός κώδικα
1	1/2 bits/symbol
2	2/3 bits/symbol
3	3/4 bits/symbol

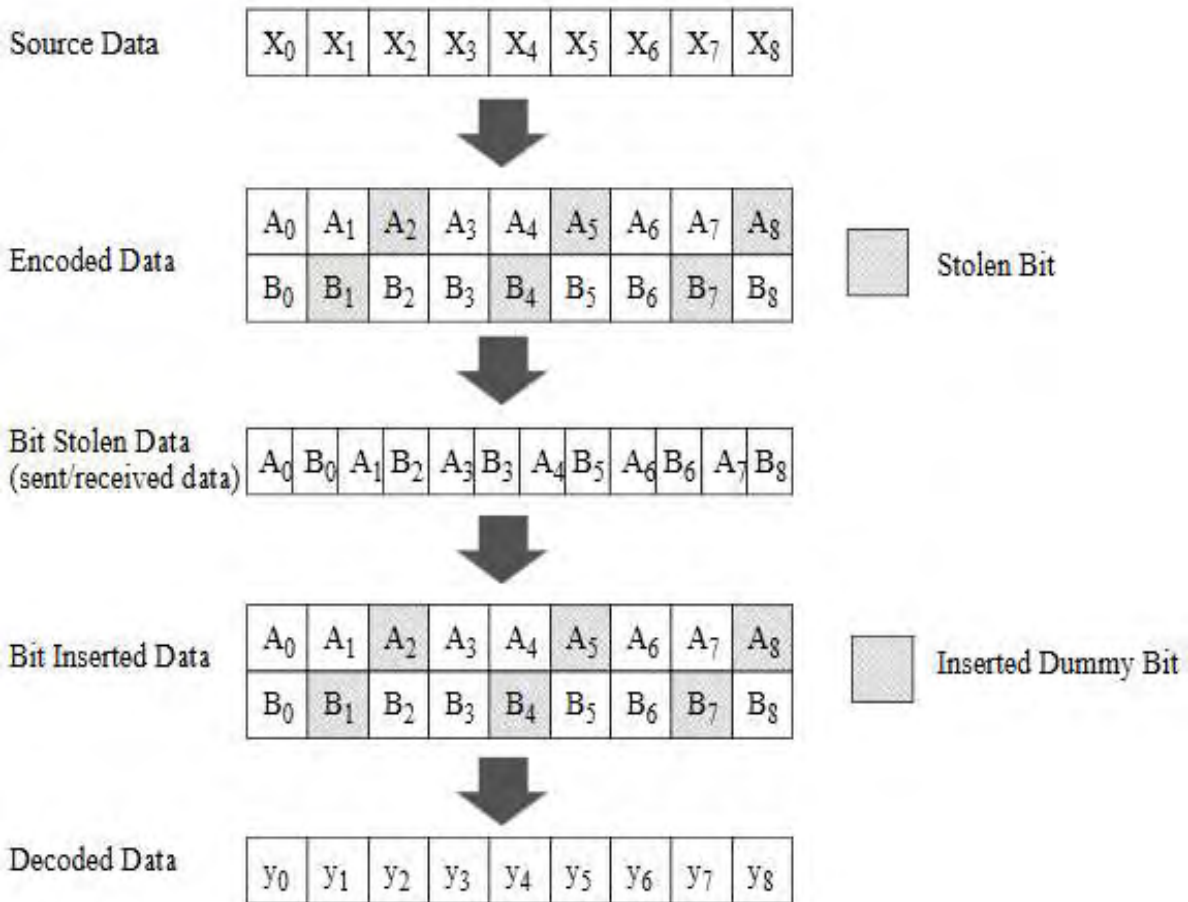
Η λειτουργία του puncture είναι απλή. Κάθε χρονική στιγμή εξάγει κάποια από τα δεδομένα που λαμβάνει. Ο τρόπος που εξάγει τα δεδομένα εξαρτάται πλήρως από τον επιθυμητό ρυθμό μετάδοσης. Το πρότυπο (pattern) εξαγωγής των δεδομένων, που ορίζει το πρωτόκολλο, φαίνεται στα σχήματα 11 και 12, για ρυθμό μετάδοσης 2/3 και 3/4, αντίστοιχα. (Τα σχήματα αυτά προέρχονται από το αρχείο του πρωτοκόλλου.) Ενώ όπως είπαμε και παραπάνω, για ρυθμό μετάδοσης 1/2, ο puncture δεν μεταβάλλει καθόλου το πλήθος των κωδικοποιημένων bit και απλά προωθεί την έξοδο του συνελκτικού κωδικοποιητή στην δική του έξοδο. Αν και δεν θα ασχοληθούμε με το τμήμα του δέκτη, αναφέρουμε ότι τα bit που αφαιρούνται κατά την εκτέλεση του puncture στην πλευρά του πομπού, συμπληρώνονται με ένα μηδενικό bit κατά την παραλαβή τους από τον δέκτη. Αυτό συμβαίνει ώστε να αποκωδικοποιηθεί σωστά το λαμβανόμενο σήμα. Η εισαγωγή των μηδενικών bit στην ακολουθία δεδομένων του λαμβανόμενου σήματος από τον δεκτή, φαίνεται και αυτή στα σχήματα 11 και 12 για κάθε έναν από τους δυνατούς ρυθμούς κώδικα. Όπως προαναφέραμε, το σύστημα του puncture θα τοποθετηθεί ακριβώς μετά τον συνελκτικό κωδικοποιητή και στην συνέχεια, θα στέλνει απευθείας το σήμα εξόδου του στην είσοδο του interleaver.

Punctured Coding ($r = 2/3$)



Σχήμα 11. Πρότυπο αφαίρεσης bit από την ακολουθία της κωδικοποιημένης πληροφορίας, για την επίτευξη ρυθμού κώδικα 2/3.

Punctured Coding ($r = 3/4$)



Σχήμα 12. Πρότυπο αφαίρεσης bit από την ακολουθία της κωδικοποιημένης πληροφορίας, για την επίτευξη ρυθμού κώδικα 3/4.

Ο συνδυασμός του puncture και του data mapper καθιστούν το σύστημα αρκετά ευέλικτο ώστε να είναι αποδοτικό σχεδόν σε κάθε κατάσταση του μέσου μετάδοσης. Όμως, αν δεν χρησιμοποιηθεί το κύκλωμα του puncture, τότε η ευελιξία που διαθέτει το σύστημα χωρίς αυτόν, δεν θα είναι αρκετή ώστε να μπορεί να προσαρμόζεται σε κάθε δυνατή κατάσταση του μέσου μετάδοσης των δεδομένων. Αυτό θα οδηγούσε, στον σχεδιασμό και την δημιουργία συνελκτικών κωδικοποιητών ειδικών για κάθε δυνατή χρήση, ώστε το σύστημα να μπορεί να ανταποκριθεί στις απαιτήσεις των χρηστών. Πράγμα που σημαίνει ότι θα αυξανόταν ο αριθμός των πρωτοκόλλων ή ότι θα αύξαναν σημαντικά την πολυπλοκότητά τους. Αντίθετα, η προσθήκη του puncture αυξάνει ελάχιστα την πολυπλοκότητα του συστήματος και λύνει το πρόβλημα προσαρμογής του. Επομένως, ο puncture με την χαμηλή πολυπλοκότητα και την ευελιξία που προσδίδει στο σύστημα επικοινωνίας, κρίνεται βασικό στοιχείο για την αύξηση της απόδοσής του.

2.3.4 Interleaver

Σε ένα κανάλι επικοινωνίας με μνήμη, τα πιθανά σφάλματα που εμφανίζονται στο μεταδιδόμενο σήμα είναι σφάλματα έκρηξης. Αν τα σφάλματα έκρηξης έχουν μέγεθος που ξεπερνούν την χωρητικότητα του συνελκτικού κωδικοποιητή, τότε αυτός αποτυγχάνει να ανακτήσει την πληροφορία. Για την αντιμετώπιση των σφαλμάτων έκρηξης χρησιμοποιείται το σύστημα του interleaver. Ο interleaver είναι το κύκλωμα που

ολοκληρώνει την διαδικασία της διόρθωσης και ανίχνευσης λαθών. Το σύστημα του interleaver λαμβάνει στην είσοδό του ένα σήμα δεδομένων, το οποίο το επεξεργάζεται με τρόπο που καθιστά δυνατή την διόρθωσή σφαλμάτων έκρηξης. Συγκεκριμένα, ανακατεύει το σήμα που λαμβάνει και μαζί με την χρήσιμη πληροφορία ανακατεύονται και τα σφάλματα. Το ανακάτεμα γίνεται με τρόπο που διασπείρει τα σφάλματα σε όλο το σύμβολο. Η διαδικασία αυτή μετατρέπει τα σφάλματα έκρηξης, που επηρεάζουν ένα σύνολο συνεχόμενων bit, σε τυχαία σφάλματα που βρίσκονται μεμονωμένα σε διάφορα σημεία μέσα στο σύμβολο.

Υπάρχουν διάφοροι τύποι interleaver, όμως το πρωτόκολλο χρησιμοποιεί έναν τμηματικό (block) interleaver. Ο τμηματικός interleaver είναι ο πιο απλός τύπος interleaver ως προς την λειτουργία του. Γράφει σε μία μνήμη με την σειρά τα δεδομένα και όταν ολοκληρωθεί η εγγραφή του τμήματος, τα διαβάζει με διαφορετική σειρά. Το γεγονός ότι πρέπει να διαβάσει ολόκληρο το τμήμα του σήματος προτού αρχίσει να ανακατεύει τα δεδομένα, αυξάνει την καθυστέρηση (delay) που προσδίδει στο σύστημα. Ένα παράδειγμα ανάγνωσης και εγγραφής στη μνήμη φαίνεται στο σχήμα 13. Στο σχήμα 13 (α) φαίνεται ο τρόπος που γίνεται η εγγραφή, αρχικά γράφεται η πρώτη σειρά, η οποία είναι σημειωμένη με πράσινο χρώμα. Ενώ στο σχήμα 13 (β) φαίνεται ο τρόπος που γίνεται η ανάγνωση των δεδομένων, πρώτα διαβάζεται η πρώτη στήλη, η οποία είναι σημειωμένη με κίτρινο χρώμα.

1	2	3	4
5	6	7	8
9	10	11	12

(α)

1	2	3	4
5	6	7	8
9	10	11	12

(β)

Σχήμα 13. (α) Σειρά εγγραφής πληροφορίας, (β) σειρά ανάγνωσης πληροφορίας, για την επίτευξη του ζητούμενου τρόπου ανακατέματος του σήματος.

Στο παράδειγμα αυτό, ο interleaver γράφει τα δεδομένα στη μνήμη, στις θέσει 1 έως 12 και τα διαβάζει με την σειρά από τις θέσεις 1, 5, 9, 2, 6, 10, 3, 7, 11, 4, 8 και 12. Για την πλήρη κατανόηση της λειτουργίας του interleaver, δίνεται στο σχήμα 14 ένα ακόμα παράδειγμα, το οποίο απεικονίζει ακριβώς τα αποτελέσματα της διαδικασίας διασποράς των σφαλμάτων έκρηξης μέσα σε ένα σήμα. Σε αυτό το παράδειγμα, το μήνυμα του πρόκειται να αποσταλεί είναι το “ααααββββγγγγ” και έχει μέγεθος δώδεκα ψηφίων. Στην πράξη ένα ψηφίο θα ήταν ένα δυαδικό bit, όμως για την καλύτερη απεικόνιση του παραδείγματος χρησιμοποιούμε γράμματα της ελληνικής αλφάβητου.

Σήμα πληροφορίας:
ααααββββγγγγ

Interleaved σήμα πληροφορίας:
αβγαβγαβγαβγ

Interleaved σήμα πληροφορίας με σφάλμα έκρηξης:
αβγα__βγαβγ

Deinterleaved σήμα πληροφορίας στον δέκτη:
αα_αβ_ββγ_γγ

Σχήμα 14. Παράδειγμα ανακατέματος του σήματος πληροφορίας και διασποράς του θορύβου έκρηξης.

Επειδή το συνολικό σύστημα απαιτείται, από το πρωτόκολλο, να είναι ευέλικτο ως προς τον ρυθμό δεδομένων, πρέπει και ο interleaver να προσαρμοστεί αναλόγως. Αυτό συμβαίνει γιατί ο interleaver είναι τμηματικός. Οπότε για κάθε δυνατή διαμόρφωση, το μέγεθος του τμήματος πρέπει να είναι ίσο με το πλήθος των κωδικοποιημένων bit που αντιστοιχούν σε ένα σύμβολο (N_{CBPS}). Το είδος της διαμόρφωσης είναι αυτό που καθορίζει το μέγεθος του N_{CBPS} . Το πλήθος των κωδικοποιημένων bit ανά σύμβολο, για κάθε δυνατό ρυθμό δεδομένων που υποστηρίζεται από το πρωτόκολλο, φαίνεται στον πίνακα 4 της ενότητας 2.1.3. Έτσι κάθε φορά, ο interleaver θα χρειάζεται να λαμβάνει υπόψην του και το διαφορετικό μέγεθος του τμήματος. Ο τμηματικός interleaver του πρωτόκολλου χρησιμοποιεί δυο επαναλήψεις, άρα αλλάζει την σειρά των δεδομένων δύο φορές με διαφορετικό τρόπο. Οι σχέσεις που περιγράφουν τις νέες θέσεις που πρόκειται να αποθηκευτούν τα bit του συμβόλου για κάθε επανάληψη, φαίνονται στις σχέσεις 6 και 7, για την πρώτη και την δεύτερη επανάληψη, αντίστοιχα. Τα δεδομένα εισόδου, τα οποία δεν έχουν υποστεί ακόμα κάποιου είδους επεξεργασία από τον interleaver, αντιστοιχούν στην μεταβλητή k της σχέσης 6. Με i αναπαρίστανται τα δεδομένα μετά την πρώτη επανάληψη. Το διάνυσμα i περιέχει, με διαφορετική σειρά, τις τιμές της εισόδου. Ενώ τα δεδομένα μετά την δεύτερη επανάληψη αποθηκεύονται στην μεταβλητή j , η οποία υπολογίζεται σύμφωνα με την σχέση 7. Ο υπολογισμός της μεταβλητής s της σχέσης 7, από την οποία εξαρτάται η δεύτερη επανάληψη, φαίνεται στην σχέση 8.

Σχέση 6

$$i = (N_{CBPS} / 16) * (k \bmod 16) + \text{Floor}(k / 16), k = 0, 1, \dots, N_{CBPS} - 1$$

Σχέση 7

$$j = s * \text{Floor}(i / s) + (i + N_{CBPS} - \text{Floor}(16 * i / N_{CBPS})) \bmod s, i = 0, 1, \dots, N_{CBPS} - 1$$

Σχέση 8

$$s = \max(N_{BPS} / 2, 1)$$

Όπως είπαμε και παραπάνω, στις σχέσεις αυτές υπάρχει ο παράγοντας N_{CBPS} ο οποίος μεταβάλλεται για κάθε διαφορετική διαμόρφωση και μαζί του μεταβάλλεται και το μέγεθος του τμήματος. Σύμφωνα με τις σχέσεις αυτές η αλλαγή του μεγέθους του τμήματος επηρεάζει και τον τρόπο ανακατέματος των δεδομένων. Συμπεραίνουμε ότι το είδος της διαμόρφωσης θα καθορίζει τον τρόπο που ανακατεύονται τα δεδομένα, αλλά και το μέγεθος του τμήματος του interleaver. Βάσει του πρωτοκόλλου η θέση του interleaver ορίζεται μετά το σύστημα του puncture και πριν το σύστημα διαμόρφωσης. Έτσι πρώτα να κωδικοποιείται το σήμα, μετά να γίνεται ανθεκτικό έναντι σφαλμάτων έκρηξης και στο τέλος να διαμορφώνεται. Το κύκλωμα διαμόρφωσης θα περιγραφεί στην επόμενη ενότητα (ενότητα 2.3.5). Αναφέρουμε ότι στην πλευρά του δέκτη θα υπάρχει το αντίστοιχο κύκλωμα που επαναφέρει τα bit δεδομένων στην αρχική τους σειρά, σύμφωνα με το πρωτόκολλο. Το σύστημα αυτό ονομάζεται deinterleaver.

Όπως είπαμε και προηγουμένως, ο interleaver χρησιμοποιείται για να προστατεύσει το σήμα δεδομένων από σφάλματα έκρηξης. Η λειτουργία του σε συνδυασμό με τη λειτουργία του συνελκτικού κωδικοποιητή, αυξάνει την συνολική ανθεκτικότητα του συστήματος σε σφάλματα. Αν δεν γινόταν χρήση ενός κυκλώματος διαχείρισης σφαλμάτων έκρηξης, το σύστημα επικοινωνίας δεν θα μπορούσε να ανταπεξέλθει μόνο με την βοήθεια του συνελκτικού κωδικοποιητή, γιατί από μόνος του δεν έχει την δυνατότητα να διορθώσει σφάλματα που ξεπερνούν σε μέγεθος το μήκος της μνήμης του. Συμπεραίνουμε ότι παρόλη την καθυστέρηση που δίνει ο interleaver στο σύστημα, είναι ένα απαραίτητο στοιχείο προκειμένου να επιτευχθεί η επικοινωνία στα περισσότερα περιβάλλοντα.

2.3.5 Σύστημα διαμόρφωσης

Ένα κανάλι που ενώνει τον πομπό και τον δέκτη ενός συστήματος επικοινωνίας, μπορεί να μεταφέρει τα δεδομένα σε συγκεκριμένες συχνότητες. Όμως οι τιμές των δεδομένων βρίσκονται, αρχικά, στη βασική ζώνη

συχνοτήτων. Οι τιμές συχνότητας βασικής ζώνης ξεκινάνε από την μηδενική συχνότητα και φτάνουν μέχρι μία μέγιστη τιμή [4]. Προκειμένου να μπορέσει να αποσταλθεί η χρήσιμη πληροφορία στο δέκτη του συστήματος, ο πομπός πρέπει να την μεταφέρει από την βασική ζώνη συχνοτήτων, στην ζώνη όπου το κανάλι επικοινωνίας μπορεί να την μεταδώσει. Η διαδικασία μεταφοράς ενός σήματος από την βασική ζώνη συχνοτήτων σε μία άλλη ζώνη, λέγεται διαμόρφωση. Η διαμόρφωση γίνεται με την βοήθεια ενός σήματος όπου η συχνότητά του είναι αυτή που απαιτεί το μέσο μετάδοσης. Το σήμα αυτό είναι γνωστό ως φέρον σήμα. Το φέρον σήμα ονομάστηκε έτσι, επειδή είναι αυτό που θα διαμορφωθεί ώστε να μεταφέρει την χρήσιμη πληροφορία. Το φέρον σήμα συνήθως είναι ημιτονοειδές και η συχνότητά του είναι πολύ μεγαλύτερη από την κεντρική συχνότητα του σήματος πληροφορίας. Όταν ο δέκτης του συστήματος παραλάβει το φέρον σήμα, το αποδιαμορφώνει προκειμένου να ανακτήσει τα δεδομένα που μεταφέρει. Η διαδικασία της αποδιαμόρφωσης είναι ακριβώς αντίστροφη της διαμόρφωσης.

Το πρωτόκολλο υποστηρίζει τέσσερα είδη διαμόρφωσης, τα οποία είναι τα BPSK, QPSK, 16-QAM και 64-QAM. Τα είδη διαμόρφωσης καθορίζουν τον ρυθμό μετάδοσης δεδομένων. Έτσι για την μετάδοση με τον ζητούμενο ρυθμό δεδομένων πρέπει να επιλεγεί η κατάλληλη διαμόρφωση. Προκειμένου τα δεδομένα να αντιστοιχηθούν σε τιμές αυτών των διαμορφώσεων, θα χρειαστεί να συλλέγονται ανά ομάδες μεγέθους N_{BPSK} bit. Τα μεγέθη του N_{BPSK} είναι 1, 2, 4 ή 6 bit, αντίστοιχα για τις τέσσερις διαμορφώσεις. Η είσοδος του συστήματος διαμόρφωσης είναι ένας ακέραιος απρόσημος αριθμός και προέρχεται από τον interleaver. Ενώ η έξοδος του παράγει μία φανταστική τιμή που την στέλνει απευθείας στην είσοδο του symbol former. Το σύστημα του symbol former είναι αυτό που λαμβάνει ένα πλήθος μεγέθους 48 φανταστικών αριθμών και τα συνενώνει σε ένα ολοκληρωμένο σύμβολο, το οποίο περιγράφεται στην ενότητα 2.3.6.

Οι φανταστικές τιμές που παράγονται ανήκουν στο πεδίο των συχνοτήτων, οι οποίες υπολογίζονται βάσει της σχέσης 9 και συμβολίζονται με το γράμμα d.

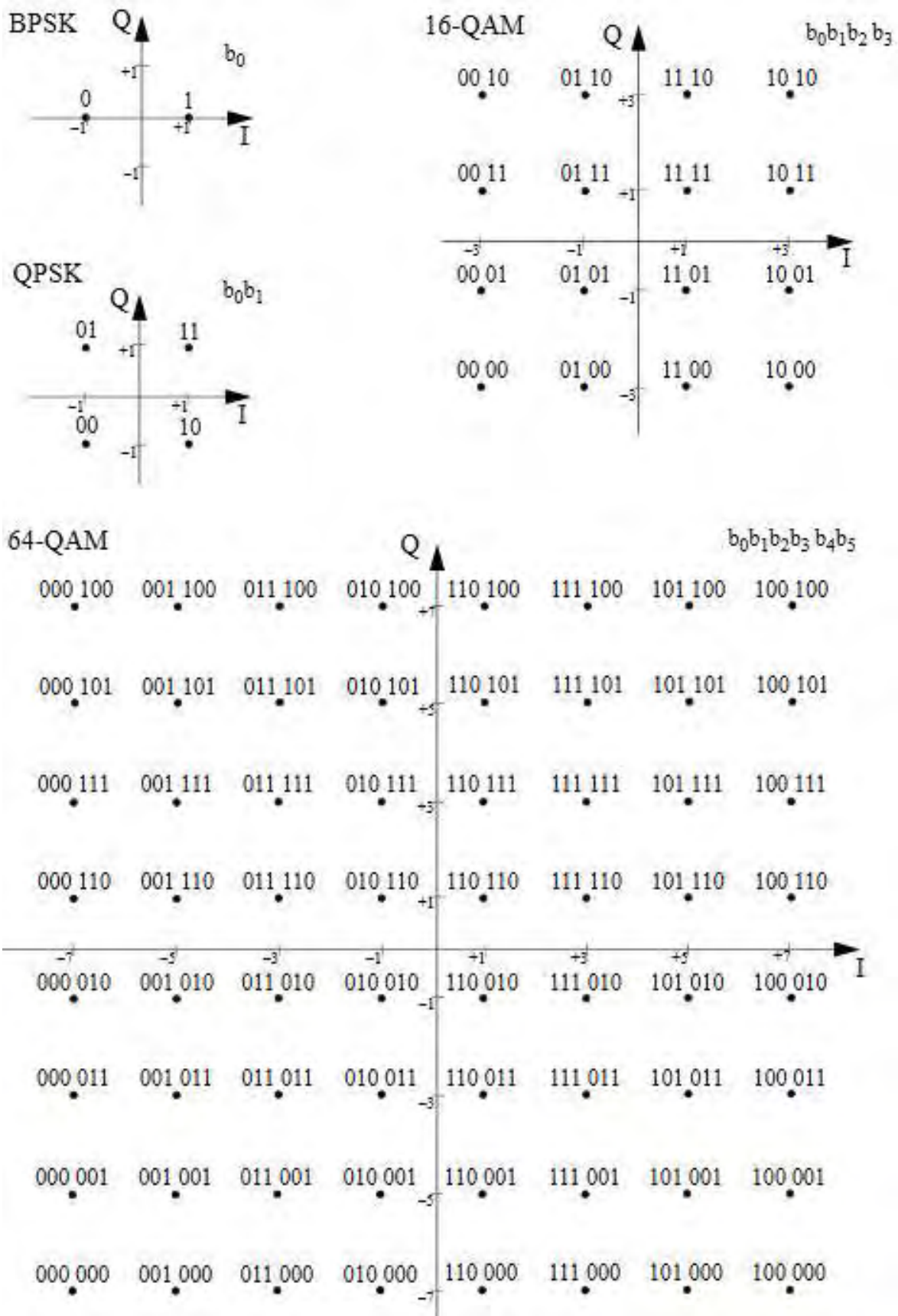
Σχέση 9

$$d = (I + jQ) * K_{MOD}$$

Οι τιμές I και Q της σχέσης 9 υπολογίζονται με την βοήθεια του Gray-code. Αυτός ο κώδικας είναι που αντιστοιχίζει τις τιμές της εισόδου με την φανταστική και την πραγματική τιμή του ζητούμενου αριθμού. Όταν υπολογισθεί ο μιγαδικός αριθμός $(I + jQ)$ πολλαπλασιάζεται με έναν παράγοντα κανονικοποίησης, ο οποίος συμβολίζεται με K_{MOD} στην σχέση 9. Στην συνέχεια στέλνεται στην έξοδο του συστήματος διαμόρφωσης. Ο παράγοντας κανονικοποίησης χρησιμοποιείται προκειμένου να επιτευχθεί ίδια μέση ισχύς για κάθε είδος διαμόρφωσης [1]. Οι πίνακες του Gray-code, που αντιστοιχίζουν τα bit δεδομένων στις μεταβλητές I και Q της σχέσης 9, φαίνονται στο σχήμα 16. Ενώ ο πίνακας με τους παράγοντες κανονικοποίησης K_{MOD} για την κάθε δυνατή διαμόρφωση είναι ο πίνακας του σχήματος 15.

Modulation	K_{MOD}
BPSK	1
QPSK	$1/\sqrt{2}$
16-QAM	$1/\sqrt{10}$
64-QAM	$1/\sqrt{42}$

Σχήμα 15. Αντιστοιχία παράγοντα κανονικοποίησης και τύπου διαμόρφωσης.



Σχήμα 16. Κώδικας Gray για τους τύπους διαμόρφωσης BPSK, QPSK, 16-QAM και 64-QAM.

2.3.6 Σύστημα συνένωσης του συμβόλου

Ο symbol former είναι το σύστημα που αναλαμβάνει να ομαδοποιήσει τα δεδομένα εισόδου σε ομάδες των 48 μιγαδικών αριθμών. Βάσει πρωτόκολλου το σύστημα διαμόρφωσης παραδίδει έναν φανταστικό αριθμό στον symbol former. Έτσι η κάθε είσοδος είναι μία τιμή στο πεδίο των συχνοτήτων. Μόλις ο symbol former συλλέξει ολόκληρο το σύνολο των 48 φανταστικών αριθμών, το ενισχύει με τέσσερις πιλοτικές (pilot) τιμές. Οι πιλοτικές τιμές είναι και αυτές μιγαδικές. Το σύστημα που τις παράγει και ο σκοπός ύπαρξής τους περιγράφονται στην ενότητα 2.3.7. Όταν δομηθεί το σύνολο των 52 φανταστικών αριθμών, που προκύπτει από τα δεδομένα και τις πιλοτικές τιμές, ο symbol former το στέλνει παράλληλα στο σύστημα του γρήγορου μετασχηματισμού Fourier προκειμένου να μετατραπεί σε τιμές στο πεδίο του χρόνου. Όμως κατά την παραλαβή τους από το σύστημα του γρήγορου μετασχηματισμού Fourier, θα χρειαστεί να αλλάξουν την σειρά τους, όπως υποδεικνύεται από το πρωτόκολλο. Το σύστημα του αντίστροφου γρήγορου μετασχηματισμού Fourier περιγράφεται στην παράγραφο 2.3.8. Το σύνολο των τιμών που θα ομαδοποιήσει ο symbol former, θα αντιστοιχηθεί στα υποφέρον σήματα του συμβόλου. Συμπεραίνουμε ότι ο symbol former πρέπει να συλλέξει ακριβώς 48 μιγαδικούς αριθμούς και να τους ενισχύσει με τις 4 πιλοτικές τιμές. Είναι σημαντικό να πούμε ότι από την στιγμή που τα δεδομένα έχουν περάσει από το στάδιο της αντιστοίχισής τους σε τιμές συχνοτήτων, δεν έχουν πλέον εξάρτηση από τον ζητούμενο ρυθμό μετάδοσης δεδομένων. Δηλαδή η διαδικασία από εδώ και στο εξής θα είναι η ίδια για κάθε ρυθμό μετάδοσης δεδομένων που απαιτεί το σύστημα.

2.3.7 Σύστημα παραγωγής πιλοτικών τιμών

Σε αυτό το σύστημα παράγονται οι 4 πιλοτικές τιμές, που θα εισαχθούν στο σύμβολο. Κάθε μία από αυτές θα αντιστοιχηθεί σε ένα από τα υποφέρον σήματα του συμβόλου. Η ενσωμάτωση των πιλοτικών τιμών μέσα στο σύμβολο γίνεται ώστε να αυξηθεί η ανθεκτικότητά του και η δυνατότητα ανίχνευσής του από τον δέκτη, έναντι στις ολισθήσεις συχνότητας και του θορύβου φάσης που μπορεί να προκύψουν. Δηλαδή οι πιλοτικές τιμές χρησιμεύουν ώστε να μπορεί να συγχρονίζεται ο δέκτης του συστήματος με το μήνυμα που παραλαμβάνει [4]. Σύμφωνα με το πρωτόκολλο αυτές οι τέσσερις πιλοτικές τιμές είναι BPSK διαμορφωμένες. Αν θεωρήσουμε ότι τα 48 υποφέρον σήματα πληροφορίας σε συνδυασμό με τα 4 υποφέρον πιλοτικά σήματα απαριθμούνται από το -26 έως το 26, τότε οι θέσεις των πιλοτικών σημάτων θα είναι οι -21, -7, 7 και 21. Ενώ οι θέσεις των σημάτων πληροφορίας θα είναι όλες οι υπόλοιπες. Οι θέσεις στις οποίες αντιστοιχίζονται όλες οι τιμές, καθορίζονται από το σύστημα του symbol former.

Το πρωτόκολλο έχει προκαθορισμένες τιμές για το κάθε πιλοτικό υποφέρον σήμα. Έτσι για τα υποφέρον σήματα -21, -7, 7 και 21 του συμβόλου, οι πιλοτικές τιμές θα είναι 1, 1, 1 και -1, αντίστοιχα. Αυτή η αντιστοιχία φαίνεται καλύτερα στην σχέση 10.

Σχέση 10

$$P_{-26, 26} = \{0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, -1, 0, 0, 0, 0, 0\}$$

Στην σχέση αυτή οι τιμές του συνόλου του δεξιού μέλους απαριθμίζονται από το -26 έως και το 26, όπως είπαμε και παραπάνω. Σε κάθε μηδενικό του συνόλου αυτού αντιστοιχεί ένα από τα 48 υποφέρον σήματα πληροφορίας. Επιπλέον, σε κάθε σύμβολο δεν ενσωματώνονται ακριβώς τα ίδια πιλοτικά σήματα. Για την διαχείριση και την ενσωμάτωση των σωστών τιμών, χρησιμοποιείται ένα δεύτερο σύνολο. Αυτό λέγεται σύνολο πολικότητας (polarity) των πιλοτικών τιμών, το οποίο φαίνεται στην σχέση 11.

Σχέση 11

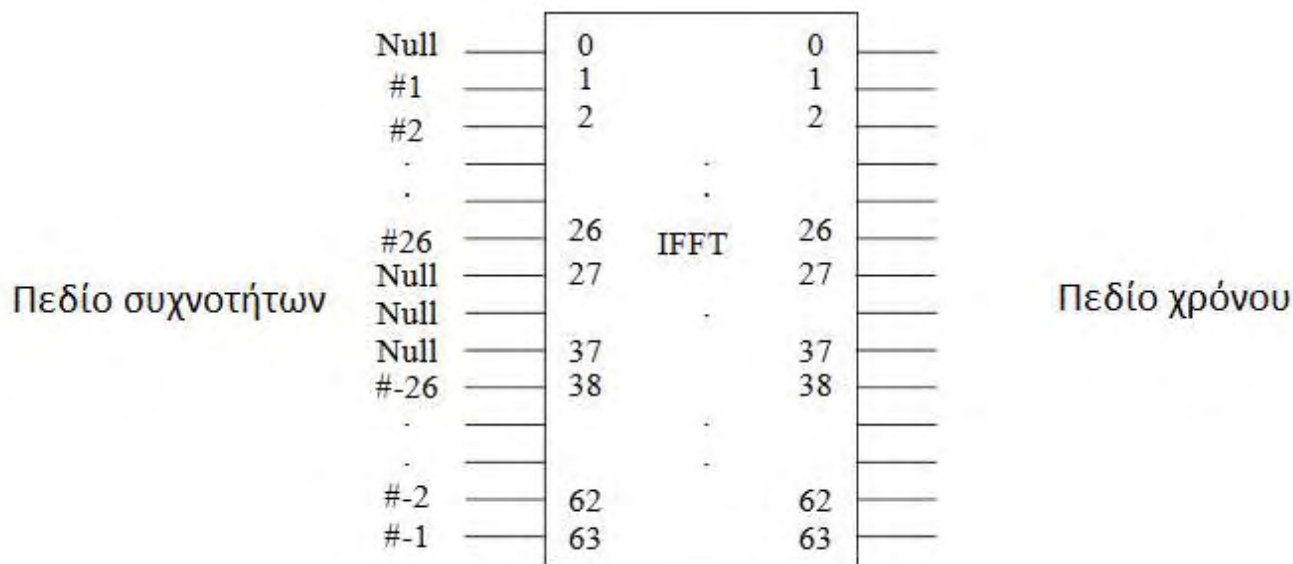
$$p_{0..126v} = \{1, 1, 1, 1, -1, -1, -1, 1, -1, -1, -1, -1, 1, 1, -1, 1, -1, -1, 1, 1, -1, 1, 1, 1, 1, 1, 1, -1, 1, 1, 1, -1, 1, 1, -1, -1, 1, 1, 1, -1, 1, -1, -1, -1, 1, -1, 1, -1, -1, 1, -1, -1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, -1, 1, -1, 1, 1, -1, -1, -1, 1, 1, -1, -1, -1, -1, 1, -1, -1, 1, -1, 1, 1, 1, 1, -1, 1, -1, 1, -1, -1, -1, 1, -1, 1, 1, 1, -1, 1, -1, 1, -1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1\}$$

Έτσι πριν την αποθήκευση των πιλοτικών τιμών μέσα στο σύμβολο, γίνεται ο πολλαπλασιασμός αυτών με την τιμή που υποδεικνύει το σύνολο πολικότητας. Το σύνολο πολικότητας υποδεικνύει με την σειρά τις τιμές ξεκινώντας από την πρώτη θέση. Έτσι δίνει την επόμενη τιμή στο κάθε νέο σύμβολο που συνενώνεται. Όταν φτάσει στο τέλος του ή όταν ολοκληρωθεί η αποστολή του πακέτου, τότε αρχίζει η εξαγωγή των τιμών πολικότητας από την αρχή.

2.3.8 Αντίστροφος γρήγορος μετασχηματισμός Fourier

“Ο μετασχηματισμός Fourier (Fourier transform (FT)) είναι το μαθηματικό εργαλείο που επιτρέπει την ανάλυση ενός σήματος σε σήματα απλών συχνοτήτων” [5]. Κατά τον μετασχηματισμό Fourier γίνεται η μετατροπή της περιγραφής του σήματος από το πεδίο του χρόνου στο πεδίο των συχνοτήτων. Για την αντίστροφη λειτουργία, όπου η περιγραφή του σήματος στο πεδίο των συχνοτήτων μεταφέρεται στο πεδίο του χρόνου, χρησιμοποιείται ο αντίστροφος μετασχηματισμός Fourier (inverse Fourier transform (IFT)). Στο σύστημα επικοινωνίας που υλοποιούμε, πριν γίνει η διαμόρφωση του μηνύματος, οι τιμές του ανήκουν στο πεδίο της βασικής ζώνης συχνοτήτων. Στην συνέχεια το σήμα του μηνύματος διαμορφώνεται και μεταβαίνει στο πεδίο συχνοτήτων που απαιτεί το κανάλι επικοινωνίας, ώστε να μπορεί να το μεταδώσει μέσω αυτού. Όμως πριν αποσταλεί το μήνυμα μέσω του καναλιού, θα πρέπει να γίνει μία ακόμα μετατροπή. Για την μετατροπή αυτή χρησιμοποιείται ο αντίστροφος μετασχηματισμός Fourier, ώστε το σήμα μηνύματος να μεταφερθεί από πεδίο των συχνοτήτων στο πεδίο του χρόνου. Αυτή η τελική μετατροπή είναι απαραίτητη, γιατί πριν την μετάδοση του μηνύματος μέσω του καναλιού, αυτό πρέπει να έρθει σε μορφή πραγματικού ηλεκτρομαγνητικού σήματος. Η παραγωγή του τελικού σήματος θα γίνει με την βοήθεια του μετατροπέα από ψηφιακό σε αναλογικό, ο οποίος στην είσοδό του απαιτεί τιμές που ανήκουν στο πεδίο του χρόνου, προκειμένου να εξάγει το επιθυμητό σήμα. Αυτός ο μετατροπέας υλοποιείται στο επίπεδο του ελέγχου πρόσβασης πολυμέσων και δεν θα ασχοληθούμε περαιτέρω μαζί του. Από την άλλη πλευρά, ο δέκτης προκειμένου να ανακτήσει τα δεδομένα, πρώτα περνάει το μήνυμα από τον μετασχηματισμό Fourier και στην συνέχεια αποδιαμορφώνεται ώστε να μεταβεί στο πεδίο βασικής ζώνης.

Η υλοποίησή μας περιλαμβάνει μόνο τον πομπό του συστήματος, οπότε καλούμαστε να σχεδιάσουμε μόνο τον αντίστροφο μετασχηματισμό Fourier που απαιτείται από το πρωτόκολλο. Έτσι στην συνέχεια της ενότητας αυτής θα ασχοληθούμε μόνο με τον αντίστροφο μετασχηματισμό Fourier. Γνωρίζουμε πως αν το σύνολο των τιμών που έχει να μετατρέψει από το πεδίο των συχνοτήτων στο πεδίο του χρόνου ένας αντίστροφος μετασχηματισμός Fourier, έχει μέγεθος που ισούται με μία δύναμη του 2, τότε αυτός μπορεί να υλοποιηθεί αποδοτικά ως αντίστροφος γρήγορος μετασχηματισμός Fourier (inverse fast Fourier transform (IFFT)). Έτσι προκειμένου το πρωτόκολλο να αυξήσει την αποδοτικότητα του συστήματος επικοινωνίας, επεκτείνει το σύνολο των 52 υποφύρον σημάτων ώστε να φτάσει να είναι ίσο με την επόμενη δυνατή δύναμη του 2 και να μπορέσει να χρησιμοποιήσει τον αντίστροφο γρήγορο μετασχηματισμό Fourier. Δηλαδή το νέο σύνολο θα έχει μέγεθος 64 υποφύρον σημάτων. Έτσι σε αυτό το σύνολο θα χωράνε τα 48 σήματα πληροφορίας και τα 4 πιλοτικά σήματα. Επιπλέον, οι υπόλοιπες 12 θέσεις θα αντιστοιχηθούν σε μηδενικά υποφύρον σήματα, ώστε να εξασφαλιστεί η προστασία από παρεμβολές γειτονικών δίαυλων [4]. Έτσι το σύστημα αντίστροφου γρήγορου μετασχηματισμού Fourier θα λαμβάνει τα σήματα δεδομένων με την σωστή σειρά, από το σύστημα symbol former, ενώ τα τοποθετεί και στις κατάλληλες εισόδους που υποδεικνύονται από το πρωτόκολλο. Στις κενές εισόδους θα αντιστοιχίζονται μηδενικές τιμές. Η αντιστοίχιση αυτή φαίνεται στο σχήμα 17 [1]. (Το σχήμα αυτό προέρχεται από το αρχείο του πρωτοκόλλου [1].)



Σχήμα 17. Σύστημα αντίστροφου γρήγορου μετασχηματισμού Fourier.

Στις εισόδους από 1 έως 26 θα αντιστοιχίζονται τα υποφέρον σήματα με την ίδια αρίθμηση, ενώ στις θέσεις από 38 έως 63 θα αντιστοιχίζονται τα υποφέρον σήματα με αρίθμηση από -26 έως -1. Στις εισόδους από 27 έως 37 και στην μηδενική (DC) είσοδο θα αντιστοιχίζονται μηδενικές τιμές, όπως αναφέραμε και παραπάνω. Η αρίθμηση των υποφέρον σημάτων έχει αναφερθεί στην ενότητα 2.3.6, όπου περιγράφεται το σύστημα του symbol former.

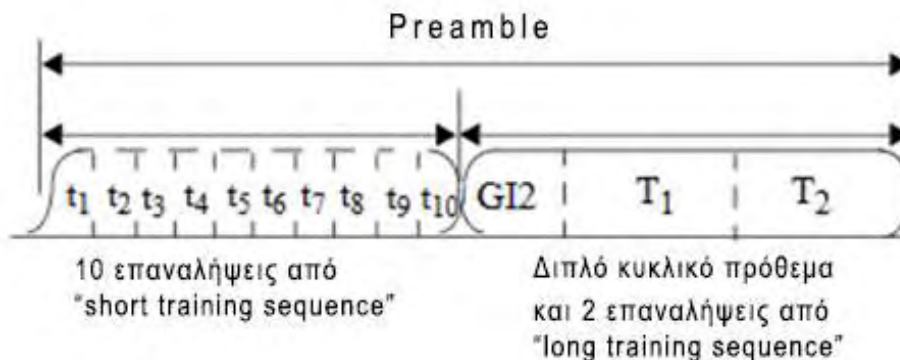
2.3.9 Εισαγωγή τιμών φύλαξης και παραθύρωσης

Κατά την παραλαβή ενός συμβόλου από τον δέκτη παρατηρείται ότι τα δεδομένα που μετέφερε έχουν αλλοιωθεί. Η αλλοίωση αυτή προκύπτει εξαιτίας του φαινομένου της διασυμβολικής παρεμβολής (intersymbol interference (ISI)) και εμφανίζεται σε μικρά τμήματα στις άκρες των συμβόλων. Η ορθογώνια πολυπλεξία με διαίρεση συχνότητας μπορεί και αποτρέπει ως ένα σημείο την διασυμβολική παρεμβολή (πέρα από την διαφερωντική παρεμβολή), όμως από μόνη της δεν είναι ικανή να εξαλείψει πλήρως αυτό το πρόβλημα. Για την ολοκληρωτική καταπολέμηση της διασυμβολικής παρεμβολής χρησιμοποιείται το σύστημα εισαγωγής κυκλικού προθέματος (cyclic prefix) [6]. Η γενική λειτουργία του συστήματος αυτού είναι η αντιγραφή του τελικού τμήματος από τις τιμές στο πεδίο του χρόνου ενός συμβόλου και η προσάρτησή του στην αρχή του ίδιου συμβόλου. Αυτό θα έχει ως αποτέλεσμα την αύξηση του όγκου της μεταδιδόμενης πληροφορίας, όμως θα λύσει το πρόβλημα της διασυμβολικής παρεμβολής.

Η διασυμβολική παρεμβολή, που εμφανίζεται κατά την παραλαβή των συμβόλων από τον δέκτη του συστήματος, έχει ένα συγκεκριμένο μέγεθος. Το μέγεθος αυτό έχει προσδιοριστεί από το πρωτόκολλο. Έτσι για την εξάλειψη της διασυμβολικής παρεμβολής το πρωτόκολλο έχει ορίσει την διαδικασία της εισαγωγής του κυκλικού προθέματος ως εξής. Το πακέτο που πρόκειται να αποσταλεί στον δέκτη απαρτίζεται από τα πεδία preamble, signal και data. Για τα δύο τελευταία πεδία έχει οριστεί το ίδιο μέγεθος κυκλικού προθέματος, ενώ στο πεδίο preamble είναι λίγο πιο σύνθετη η εισαγωγή του προθέματος αυτού. Στο τμήμα των short OFDM training symbol του preamble δεν θα εισαχθεί κάποιο πρόθεμα, όμως στο τμήμα των long OFDM training symbol θα εισαχθεί ένα πρόθεμα που θα έχει το διπλάσιο μέγεθος από αυτό του προθέματος που θα εισαχθεί στα πεδία signal και data. Πιο συγκεκριμένα το μέγεθος του κυκλικού προθέματος στα σύμβολα των πεδίων signal και data είναι ίσο με 16 τιμές. Από την άλλη πλευρά, το μέγεθος του κυκλικού προθέματος στο σύμβολο του τμήματος short OFDM training symbol είναι ίσο με 32 τιμές. Έτσι από το κάθε σύμβολο θα αντιγράφονται, κατά περίπτωση, οι 16 ή οι 32 τελευταίες τιμές και θα προσαρτώνται στην αρχή του. Η λειτουργία εισαγωγής

κυκλικού προθέματος ονομάζεται Guard interval (GI). Πρέπει να πούμε ότι η λειτουργία του συστήματος που περιγράφεται δεν είναι μόνο αυτή που αναφέρθηκε. Εκτός από τα παραπάνω, το σύστημα αυτό αφού εισαγάγει το προβλεπόμενο κυκλικό πρόθεμα, υποδιπλασιάζει την πρώτη τιμή του επεκταμένου συμβόλου και την αυξάνει κατά μία δεύτερη τιμή που έχει κρατήσει από την επεξεργασία προηγούμενου συμβόλου, προτού αποσταλθεί στο δέκτη του συστήματος επικοινωνίας. Η δεύτερη τιμή, βάση της οποίας γίνεται η αύξηση, είναι η υποδιπλασιασμένη πρώτη τιμή του προηγούμενου συμβόλου χωρίς το κυκλικό πρόθεμα. Η λειτουργία υποδιπλασιασμού των τιμών αυτών ονομάζεται windowing. Το windowing εξομαλύνει τις ακραίες τιμές ενός συμβόλου ώστε να μην υπάρχουν απότομες αλλαγές στην ενέργειά του σήματος.

Όπως είπαμε και στην ενότητα 2.1.1, το πεδίο preamble χρησιμοποιείται ώστε να συγχρονιστεί ο δέκτης του συστήματος με το λαμβανόμενο μήνυμα. Για την αποτελεσματικότητα της χρήσης του απαιτείται μία επιπλέον επέκτασή του. Έτσι τα σύμβολα, προτού εισαχθεί το πρόθεμα, των δύο τμημάτων του preamble θα χρειαστεί να επεκταθούν λίγο ακόμα. Πιο συγκεκριμένα, οι τιμές στο πρώτο τμήμα του preamble επαναλαμβάνονται ανά 16 και η επέκταση που θα γίνει είναι να εξαχθούν 10 ομάδες αυτών των 16 αριθμών. Έτσι το πρώτο τμήμα θα καταλήξει να έχει συνολικά 160 τιμές. Από την άλλη πλευρά, στο δεύτερο τμήμα του preamble θα τοποθετηθεί το πρόθεμα μεγέθους 32 τιμών και στην συνέχεια θα εξαχθούν δύο φορές οι 64 τιμές του συμβόλου του. Έτσι και το δεύτερο τμήμα θα φτάσει να έχει 160 τιμές. Η τελική μορφή του preamble, που θα αποσταλθεί μέσω του καναλιού επικοινωνίας στον δέκτη του συστήματος, φαίνεται στο σχήμα 18.



Σχήμα 18. Δομή του Preamble στην τελική του μορφή.

Η τελευταία λειτουργία που επεκτείνει το μέγεθος των τμημάτων του preamble αποτελεί το μέρος του συστήματος GI και windowing. Όπως παρατηρείται οι τρεις λειτουργίες, που περιγράψαμε, θα ενσωματώνονται στο ίδιο σύστημα.

Συνοψίζουμε λέγοντας ότι το σύστημα GI και windowing θα απαρτίζεται από την λειτουργία εισαγωγής κυκλικού προθέματος, υποδιπλασιασμού συγκεκριμένων τιμών και επέκτασης των τιμών του προθέματος του πακέτου δεδομένων. Η θέση του συστήματος GI και Windowing είναι ακριβώς μετά τον αντίστροφο γρήγορο μετασχηματισμό Fourier, ώστε να προσαρτεί το κυκλικό πρόθεμα και τις τιμές φύλαξης όσο οι τιμές του συμβόλου βρίσκονται στο πεδίο του χρόνου. Ενώ η έξοδος του αποστέλνεται απευθείας στο επόμενο επίπεδο του πομπού, το οποίο είναι το επίπεδο εξάρτησης φυσικού μέσου (Physical Medium Dependent (PMD)). Το επίπεδο ελέγχου πρόσβασης πολυμέσων δεν αποτελεί τμήμα της παρούσας εργασίας.

3 FPGA

Μία συστοιχία πυλών προγραμματιζόμενου πεδίου (Field Programmable Gate Arrays (FPGA)) είναι μία συσκευή ημιαγωγών που βασίζεται σε ένα δομημένο πεδίο επαναρυθμιζόμενων λογικών block (Configurable logic blocks (CLBs)) που συνδέονται μεταξύ τους μέσω επαναρυθμιζόμενων διασυνδέσεων. Επιπλέον, ένα FPGA περιέχει και blocks εισόδου/εξόδου, τα οποία επιτρέπουν την επικοινωνία του FPGA με το υπόλοιπο

σύστημα. Τα FPGAs μπορούν και προγραμματίζονται ώστε να υλοποιούν την επιθυμητή λειτουργία, από τον χρήστη. Για τον προγραμματισμό των FPGAs χρησιμοποιούνται γλώσσες περιγραφής υλικού (Hardware description language (HDL)). Οι δύο πιο δημοφιλείς γλώσσες περιγραφής υλικού είναι η VHDL και η Verilog. Η σχεδίαση στην επιθυμητή γλώσσα θα χρειαστεί να συνθεθεί και στην συνέχεια να παραχθεί το κατάλληλο bitstream αρχείο, βάση του οποίου το FPGA επαναρυθμίζεται. Με την ολοκλήρωση της ρύθμισης του FPGA υλοποιείται και η ζητούμενη λειτουργία, η οποία είναι έτοιμη για χρήση. Η ροή σχεδίασης για τον επιτυχή επαναπρογραμματισμό ενός FPGA θα περιγραφεί στην συνέχεια. Η δομή ενός FPGA φαίνεται στο σχήμα 19.

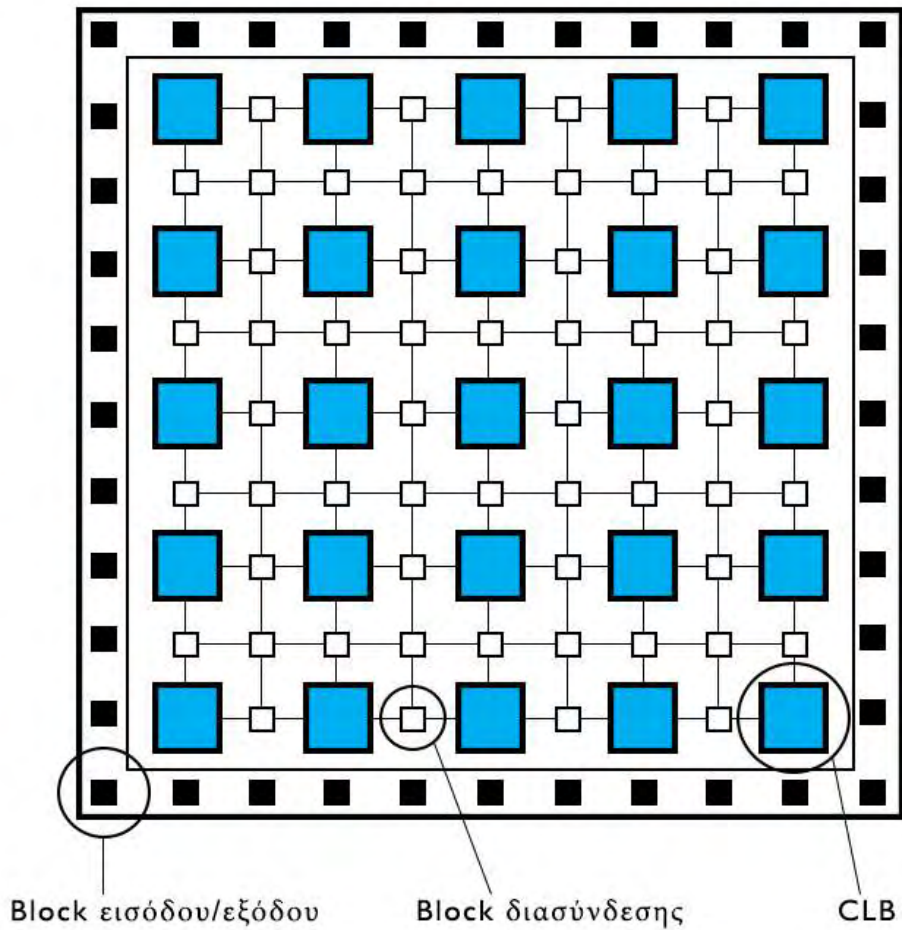
Σε χαμηλότερο επίπεδο, τα CLBs απαρτίζονται από έναν look-up table (LUT), ένα D-flip flop και έναν πολυπλέκτη. Ένα LUT υλοποιεί ένα πίνακα αληθείας. Συγκεκριμένα χρησιμοποιεί ένα σύνολο λογικών πυλών ώστε να υλοποιήσει τον λογικό πίνακα αληθείας που ορίζει το bitstream αρχείο για αυτό. Τα LUTs υλοποιούν μόνο συνδυαστικά κυκλώματα χωρίς ανατροφοδότηση. Το μέγεθος των πινάκων αληθείας των LUTs εξαρτάται από το πλήθος των εισόδων τους. Το μέγεθος των πινάκων αληθείας χαρακτηρίζεται από την σχέση 12.

Σχέση 12

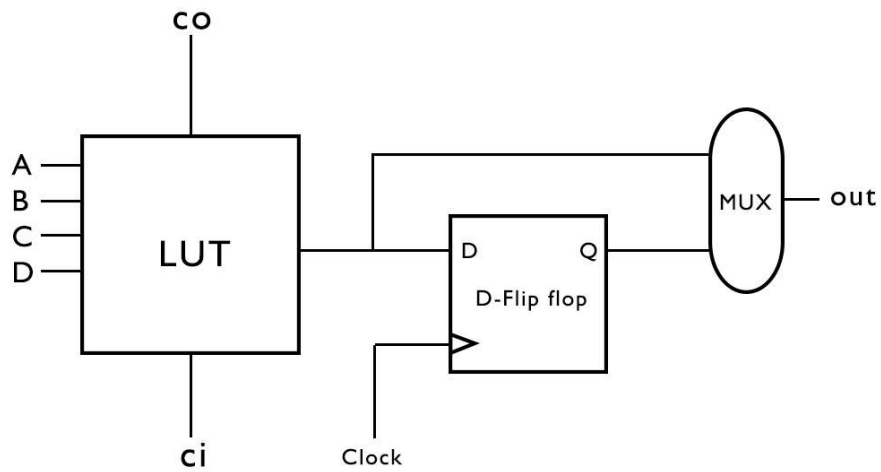
Μέγεθος πινάκων αληθείας = 2^N , όπου N είναι το πλήθος των εισόδων.

Συνήθως στα FPGA τα CLBs διαθέτουν 4 εισόδους συν την είσοδο του ρολογιού. Το σχηματικό διάγραμμα ενός CLB με 4 εισόδους φαίνεται στο σχήμα 20. Όλα τα CLBs ενός FPGA έχουν την ίδια δομή. Σημειώνουμε, ότι τα σήματα carry in (ci) και carry out (co) συνδέουν τα γειτονικά μόνο CLBs και χρησιμοποιούνται για την αποδοτική υλοποίηση αθροιστών και μετρητών.

FPGA



Σχήμα 19. Δομή ενός FPGA.



Σχήμα 20. Δομή προγραμματιζόμενου λογικού block.

Για την κατανόηση των LUTs δίνεται ένα παράδειγμα. Έστω ότι πρέπει να υλοποιηθεί η λειτουργία της λογικής συνάρτησης που φαίνεται στην σχέση 13.

Σχέση 13

$$\text{Out} = A * B + C * D$$

Για την υλοποίηση της σχέσης δεν θα χρησιμοποιηθούν οι λογικές πύλες που θα έδιναν το σωστό αποτέλεσμα, αλλά ο λογικός πίνακας που περιγράφει την σχέση αυτή θα αντιγραφεί στο LUT. Έτσι για την σχέση 13, το αντίστοιχο LUT θα είναι αυτό του πίνακα 6.

Πίνακας 6. Παραγόμενος πίνακας αληθείας για την σχέση 13.

A	B	C	D	Out
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

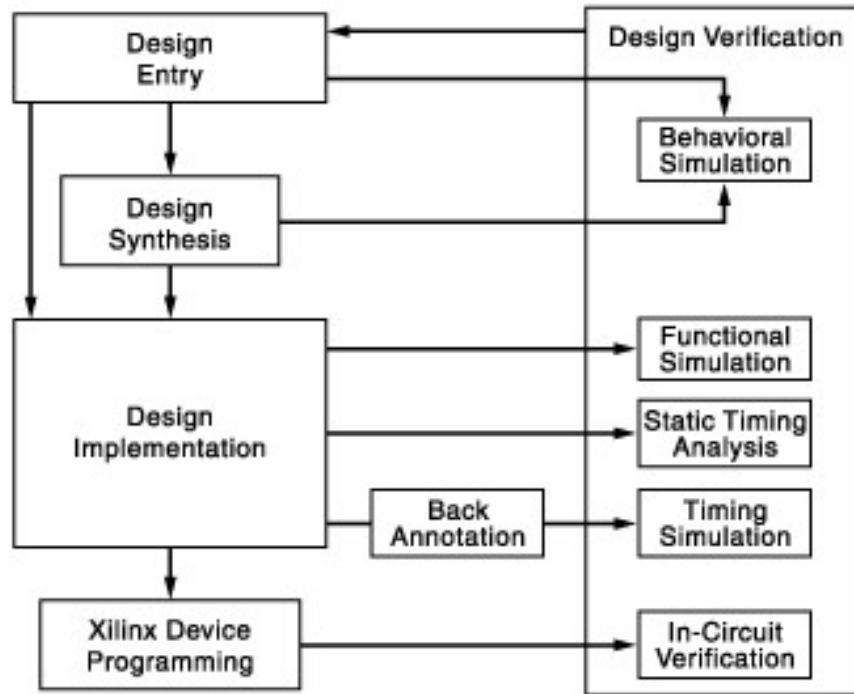
Εφόσον για οποιονδήποτε συνδυασμό εισόδων το αποτέλεσμα είναι γνωστό, τότε όποια είσοδος και να εισαχθεί η έξοδος θα προκύψει από την αντίστοιχη θέση του LUT. Με αυτόν τον τρόπο υλοποιείται κάθε συνδυαστική λογική μέσα σε ένα FPGA.

Το μεγάλο πλήθος των CLB επιτρέπουν την υλοποίηση μεγάλων και πολύπλοκων σχεδιάσεων. Όπως τα CLB έτσι και οι διασυνδέσεις αυτών είναι προγραμματιζόμενες. Έτσι βάση του bitstream αρχείου, κάθε μία από αυτές αλλάζει κατάσταση και διασυνδέει τα CLB με τρόπο που καθιστά δυνατή την υλοποίηση της σχεδίασης. Ενώ τα block εισόδου/εξόδου χρησιμοποιούνται είτε για να εισαχθούν διάφορα λειτουργικά σήματα εντός του FPGA, όπως το σήμα του ρολογιού είτε για την εκμετάλλευσή τους από τους χρήστες. Τα pin που διατίθενται στους χρήστες ονομάζονται pins γενικού σκοπού (general purpose (GP)).

3.1 Ροή σχεδίασης στο FPGA

Η ροή σχεδίασης σε ένα FPGA αποτελείται από 5 βασικά βήματα. Τα βήματα αυτά φαίνονται στο διάγραμμα ροής του σχήματος 21, όπου περιγράφεται η ροή σχεδίασης όπως ορίζεται από την Xilinx [7]. Τα βήματα που απαρτίζουν την διαδικασία αυτή είναι η σχεδίαση του κυκλώματος και η εισαγωγή των περιορισμών που αντιστοιχούν σε αυτό, η σύνθεση, η υλοποίηση, ο προγραμματισμός του FPGA και η τελική επαλήθευση του κυκλώματος που υλοποιήθηκε. Κάθε ένα από τα πρώτα 4 βήματα συνοδεύεται από το αντίστοιχο βήμα προσομοίωσης και επαλήθευσης, όπως ακριβώς φαίνεται και στο σχήμα. Αν οποιοδήποτε από

τα βήματα προσομοίωσης και επαλήθευσης αποτύχει να ολοκληρωθεί, τότε η διαδικασία επαναφέρεται στο αρχικό βήμα.



Σχήμα 21. Ροή σχεδίασης στο FPGA σύμφωνα με την Xilinx.

Στο πρώτο βήμα πρέπει να γραφτεί ο κώδικας σε μία γλώσσα περιγραφής υλικού, ο οποίος θα περιγράψει το κύκλωμα που ζητείται να υλοποιηθεί στο FPGA. Στο ίδιο βήμα πρέπει να δοθεί και ένα αρχείο όπου θα περιγράφονται οι περιορισμοί της σχεδίασης. Για παράδειγμα, ένας περιορισμός σχεδίασης είναι η συχνότητα του ρολογιού που χρησιμοποιεί το σύστημα ή οι αντιστοιχίσεις των pin γενικού σκοπού με σήματα εκτός του FPGA. Όταν ολοκληρωθεί αυτό το βήμα, η σχεδίαση προσομοιώνεται και εάν δεν λειτουργεί επιθυμητά, τότε επανεξετάζεται και διορθώνεται η σχεδίαση του κυκλώματος. Δηλαδή η διαδικασία σχεδίασης ξεκινάει από την αρχή. Αν η προσομοιωμένη σχεδίαση λειτουργεί επιθυμητά, τότε ξεκινάει η σύνθεση.

Κατά την διάρκεια της σύνθεσης ελέγχεται η σωστή σύνταξη του κώδικα και οι απαιτήσεις σε πόρους της σχεδίασης. Επιπλέον, ο χρήστης πρέπει να ελέγξει την ομοιότητα του σχηματικού διαγράμματος που σχεδίασε με αυτό που προκύπτει μετά την σύνθεση. Όπως και πριν, αν ο έλεγχος ολοκληρωθεί επιτυχώς, τότε η διαδικασία σχεδίασης θα πάει στο επόμενο βήμα, αλλιώς θα αρχίσει το ίδιο βήμα από την αρχή. Το επόμενο βήμα είναι η υλοποίηση της σχεδίασης. Σε αυτό το βήμα ελέγχονται οι περιορισμοί που δόθηκαν και εάν ικανοποιούνται ο κώδικας της σχεδίασης μεταφράζεται σε ένα bitstream αρχείο, το οποίο καθορίζει την λειτουργία των CLBs του FPGA και ποια ακριβώς θα χρησιμοποιηθούν από όλο το σύνολο αυτών. Όταν ολοκληρωθεί αυτό το βήμα, ο χρήστης μπορεί να ελέγξει εάν τον ικανοποιεί η ο τρόπος με τον οποίο έγινε η υλοποίηση. Σε κάθε περίπτωση ο χρήστης μπορεί να ρυθμίσει τον τρόπο που γίνεται η υλοποίηση ώστε να επιτευχθεί το ζητούμενο αποτέλεσμα.

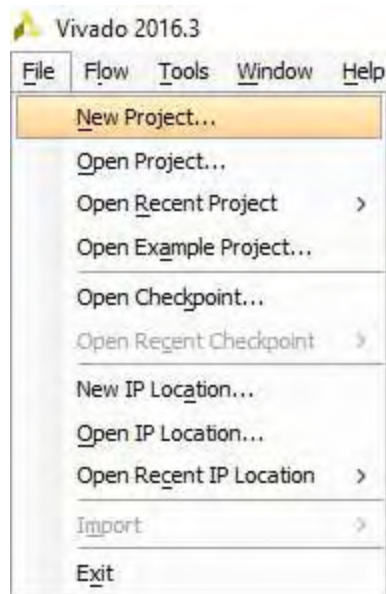
Το επόμενο βήμα εκτελείται αυτοματοποιημένα από το εργαλείο σχεδίασης. Στο βήμα αυτό γίνεται ο προγραμματισμός του FPGA. Για τον προγραμματισμό του FPGA χρησιμοποιείται το bitstream αρχείο που παράχθηκε στο βήμα της υλοποίησης. Όταν ολοκληρωθεί ο προγραμματισμός του FPGA, εκτελείται και ο τελικός έλεγχος της λειτουργία του κυκλώματος που υλοποιήθηκε. Ο έλεγχος αυτός μπορεί να γίνει χειρωνακτικά από τον χρήστη (για μικρές σχεδιάσεις), αυτόματα με την βοήθεια soft processor που ενσωματώνουν κάποιες πλακέτες, ή με την βοήθεια λογισμικών που εκτελούνται στον υπολογιστή.

3.2 Ροή σχεδίασης στο Vivado της Xilinx

Η ροή σχεδίασης που ακολουθήσαμε είναι αυτή που αναφέρουμε στην προηγούμενη ενότητα (ενότητα 3.1). Τα βήματα αυτά θα εκτελεστούν στο εργαλείο σχεδίασης Vivado, το οποίο ανήκει στην Xilinx και θα δοθούν στιγμιότυπα για την ανάδειξή τους. Το πρώτο βήμα είναι η σχεδίαση του κυκλώματος. Για την σχεδίαση αυτή χρησιμοποιήσαμε την γλώσσα περιγραφής υλικού VHDL. Το κύκλωμα που σχεδιάσαμε είναι το PLCP επίπεδο που ορίζει το πρωτόκολλο IEEE 802.11a για τον πομπό του συστήματος επικοινωνίας. Η λειτουργία και η χρησιμότητα των συστημάτων που περιέχει περιγράφονται στην ενότητα 2.3. Οι σχεδιάσεις των συστημάτων αυτών περιγράφονται στην ενότητα 4. Ενώ στην ενότητα 5 περιγράφουμε τον τρόπο με τον οποίο έγινε η προσομοίωση και η επαλήθευση της σχεδίασης των κυκλωμάτων.

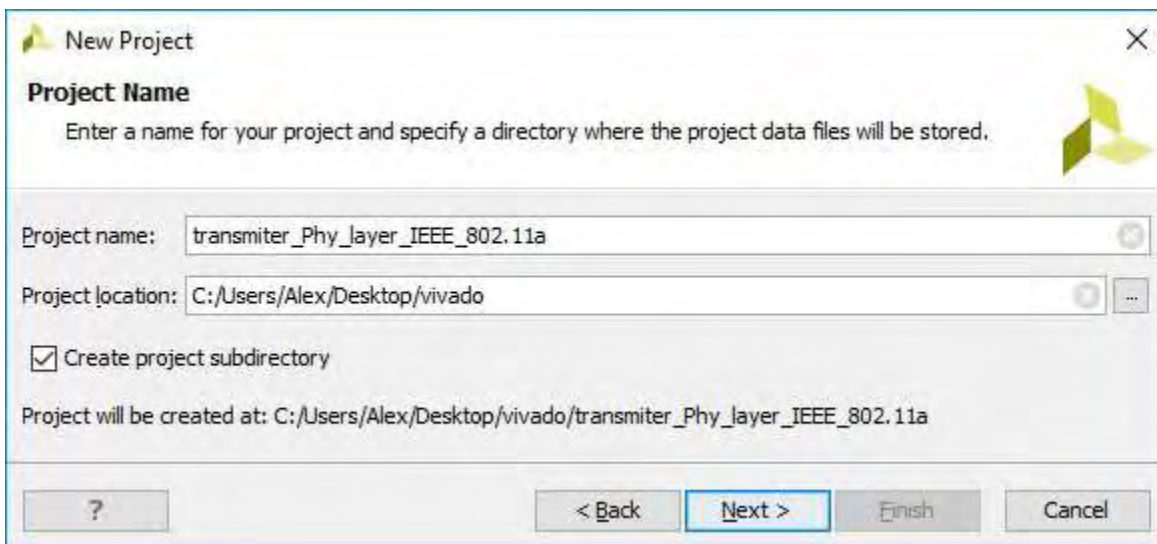
3.2.1 Δημιουργία project και εισαγωγή αρχείων σχεδίασης

Αρχικά εκκινούμε το Vivado και στην συνέχεια δημιουργούμε ένα νέο project. Για να γίνει αυτό πρέπει να επιλέξουμε από το μενού το File -> New Project, όπως φαίνεται στο σχήμα 22.



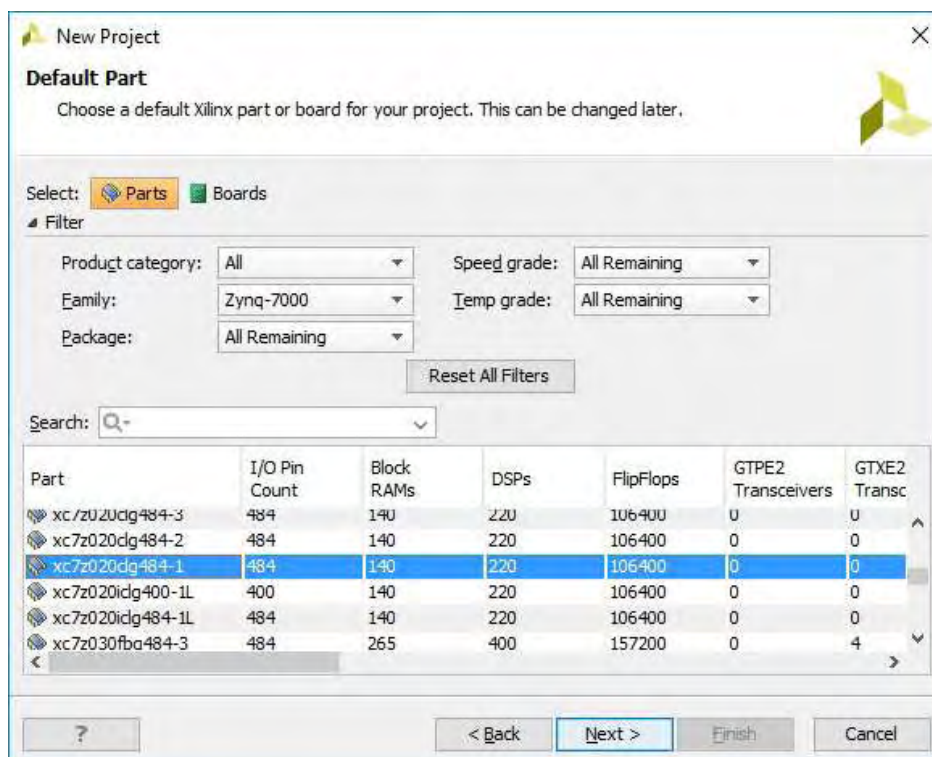
Σχήμα 22. Δημιουργία νέου project.

Στο παράθυρο που ανοίγει δίνουμε τα στοιχεία που χρειάζονται για την δημιουργία του project, όπως το όνομα του project και τον φάκελο που αυτό θα περιέχεται. Το παράθυρο αυτό φαίνεται στο σχήμα 23.



Σχήμα 23. Εισαγωγή στοιχείων του νέου project.

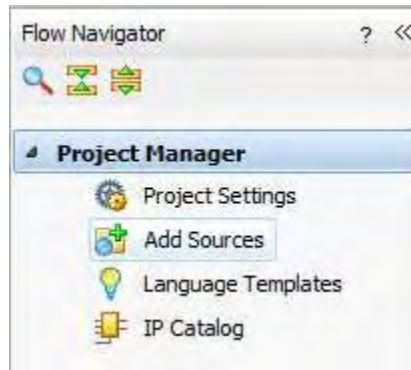
Πατάμε το κουμπί “Next” και προσπερνάμε τα παράθυρα “Add sources” και “Add constraints”. Τα αρχεία και τους περιορισμούς της σχεδίασης θα τους προσθέσουμε στην συνέχεια. Όταν φτάσουμε στο παράθυρο που λέγεται “Default part”, προσδιορίζουμε το μοντέλο FPGA που χρησιμοποιήσαμε. Εμείς χρησιμοποιήσαμε το xc7z020clg484-1, όπως φαίνεται και στο σχήμα 24.



Σχήμα 24. Επιλογή του FPGA.

Πατάμε “Next” σε αυτό και στα επόμενα παράθυρα και στο τέλος πατάμε το κουμπί “Finish” για την ολοκλήρωση της δημιουργίας του project. Πλέον, χρειάζεται να εισαγάγουμε τα αρχεία της σχεδίασης. Για να

γίνει αυτό πρέπει να πατήσουμε στο κουμπί “Add sources” του Flow Navigator. Το κουμπί αυτό βρίσκεται στα αριστερά της οθόνης και φαίνεται στο σχήμα 25.



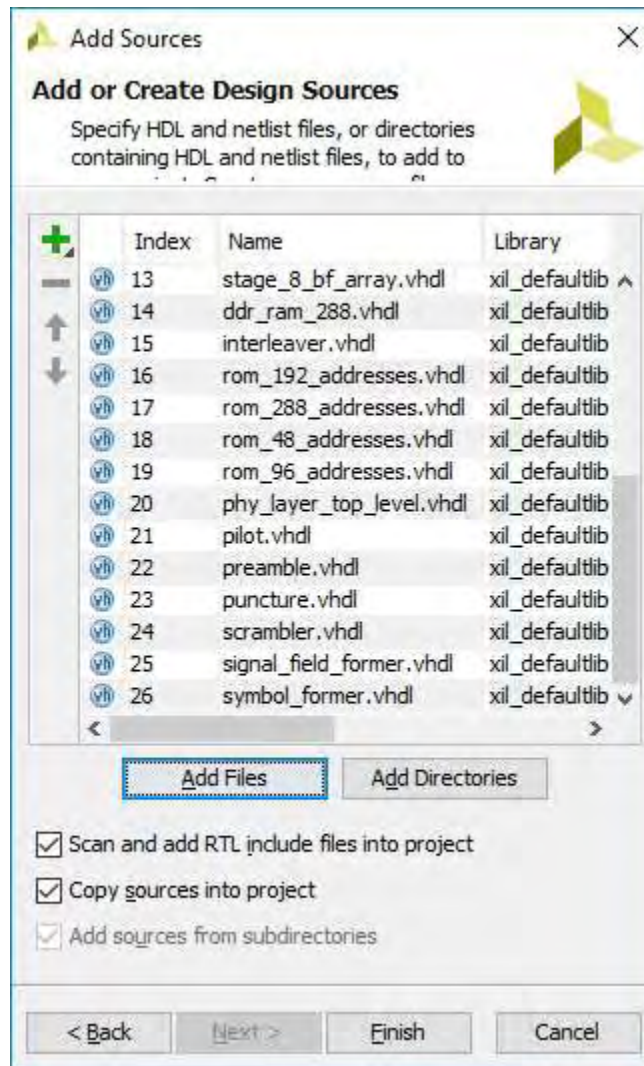
Σχήμα 25. Κουμπί για την εισαγωγή αρχείων στην σχεδίαση.

Θα εμφανιστεί ένα παράθυρο με το ίδιο όνομα. Στο παράθυρο αυτό επιλέγουμε “Add or create design sources”, όπως φαίνεται στο σχήμα 26, ώστε να μπορέσουμε να προσθέσουμε τις σχεδιάσεις των κυκλωμάτων στο project.



Σχήμα 26. Επιλογή εισαγωγής αρχείων σχεδίασης.

Θα εμφανιστεί ένα νέο παράθυρο, με το οποίο θα εισάγουμε τα αρχεία. Στο παράθυρο αυτό πατάμε το κουμπί “Add files”, για να προσθέσουμε όλα τα αρχεία των σχεδιάσεων που χρειάζονται. Μετά την εισαγωγή των αρχείων το παράθυρο θα έχει την μορφή που φαίνεται στο σχήμα 27.

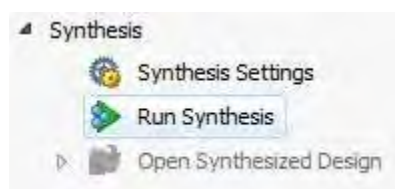


Σχήμα 27. Επιλογή των αρχείων της σχεδίασης του φυσικού επιπέδου του πομπού του πρωτοκόλλου IEEE 802.11a.

Για να ολοκληρωθεί η διαδικασία, πατάμε στο κουμπί “Finish”. Για την ολοκλήρωση του πρώτου βήματος σχεδίασης θα πρέπει να εισαγάγουμε και τον περιορισμό του ρολογιού. Όμως, επειδή θέλουμε να προσθέσουμε τον περιορισμό του ρολογιού με την χρήση ενός wizard παραθύρου, θα πρέπει να συνθέσουμε πρώτα την σχεδίαση.

3.2.2 Σύνθεση σχεδίασης

Στο δεύτερο βήμα σχεδίασης, για να γίνει η σύνθεση των κυκλωμάτων πατάμε στο κουμπί “Run Synthesis” στο Flow Navigation. Το κουμπί για την εκκίνηση της σύνθεσης φαίνεται στο σχήμα 28.



Σχήμα 28. Κουμπί για την έναρξη της σύνθεσης της σχεδίασης.

Στο παράθυρο που εμφανίζεται πατάμε “OK”, ώστε να εκτελεστεί η σύνθεση με τις προεπιλεγμένες ρυθμίσεις. Όταν ολοκληρωθεί η σύνθεση της σχεδίασης, ολοκληρώνεται και το δεύτερο βήμα της ροής σχεδίασης.

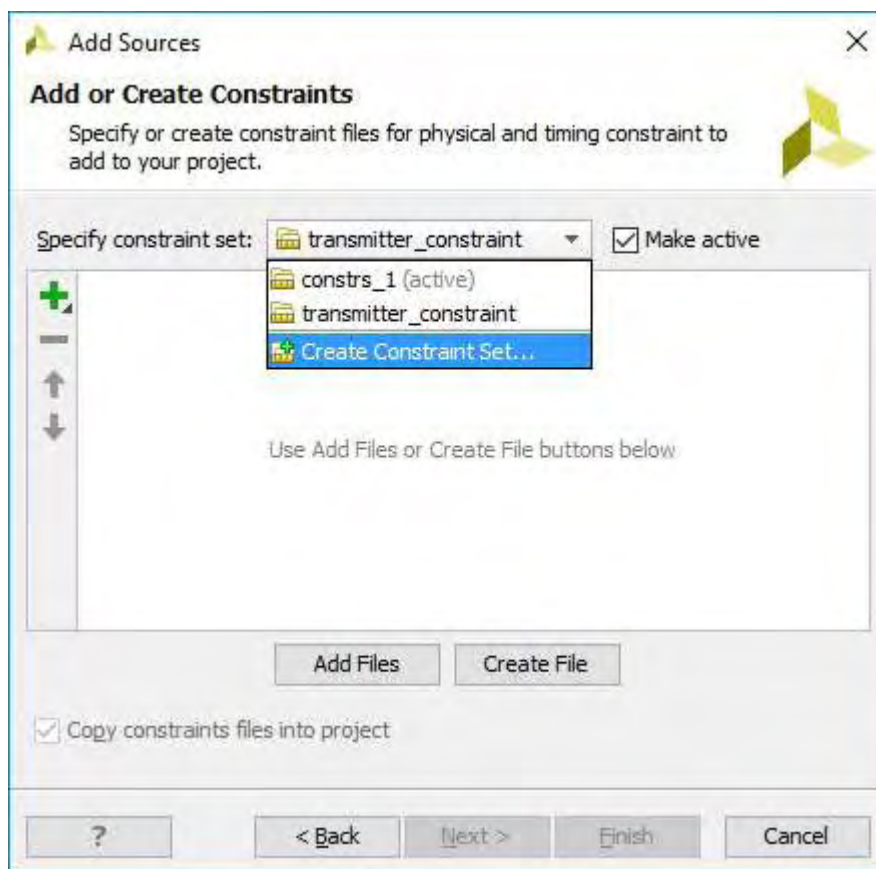
3.2.3 Δημιουργία του αρχείου περιορισμού

Τώρα πρέπει να εισαχθεί ο περιορισμός του ρολογιού. Το Vivado διαθέτει ένα εύκολο σύστημα για την εισαγωγή των περιορισμών του ρολογιού. Στην περίπτωση μας, ο μοναδικός περιορισμός που θα εισάγουμε είναι η περίοδος του ρολογιού. Η περίοδος που θα εισάγουμε είναι 90 MHz. Στην ενότητα 3.3, όπου δίνονται οι αναφορές της σχεδίασης, φαίνεται και η τήρηση του περιορισμού αυτού. Για την εισαγωγή των περιορισμών πρέπει να φτιάξουμε πρώτα το αρχείο που τους περιέχει. Για να γίνει αυτό, πατάμε στο κουμπί “Add Sources” του Flow Navigator, όπως είδαμε και παραπάνω στο σχήμα 25. Στο παράθυρο που θα εμφανιστεί (σχήμα 29), επιλέγουμε το “Add or create constraints” και πατάμε “Next”.



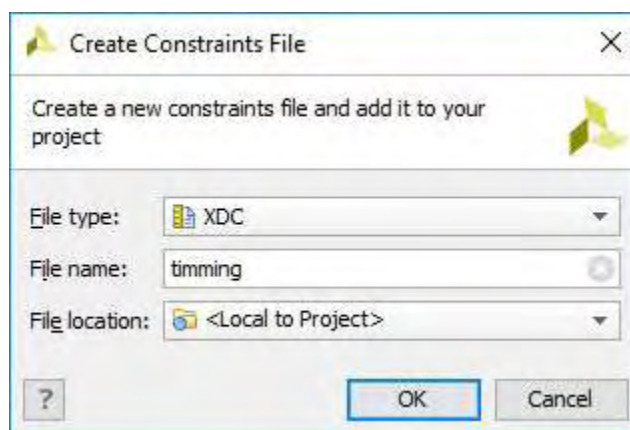
Σχήμα 29. Επιλογή εισαγωγής αρχείων περιορισμού στην σχεδίαση.

Στην συνέχεια, στο καινούργιο παράθυρο με τον ίδιο τίτλο, επιλέγουμε το “Make active” και “Create constraint set”, όπως φαίνεται και στο σχήμα 30.



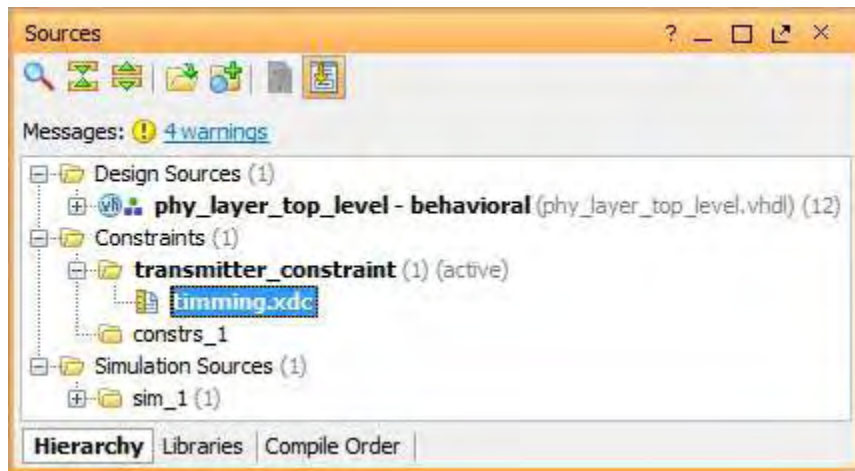
Σχήμα 30. Δημιουργία νέου αρχείου περιορισμών.

Στο επόμενο παράθυρο, δίνουμε το όνομα των περιορισμών. Εμείς το ονομάζουμε “timing” μιας και μιλάμε για χρονικούς περιορισμούς. Επιπλέον, στο πεδίο “File location” επιλέγουμε το “<Local to Project>”, ώστε το αρχείο περιορισμών να αποθηκευτεί εντός του φακέλου που υπάρχει το project. Το νέο αυτό παράθυρο φαίνεται στο σχήμα 31.



Σχήμα 31. Εισαγωγή στοιχείων για την παραγωγή του αρχείου περιορισμών.

Για να ολοκληρωθεί η δημιουργία του αρχείου περιορισμών πατάμε “OK” ή “Finish” στα παράθυρα που έχουν ανοίξει. Όπως φαίνεται και στο σχήμα 32 το αρχείο περιορισμών δημιουργήθηκε επιτυχώς.



Σχήμα 32. Επιτυχία παραγωγής του αρχείου περιορισμών.

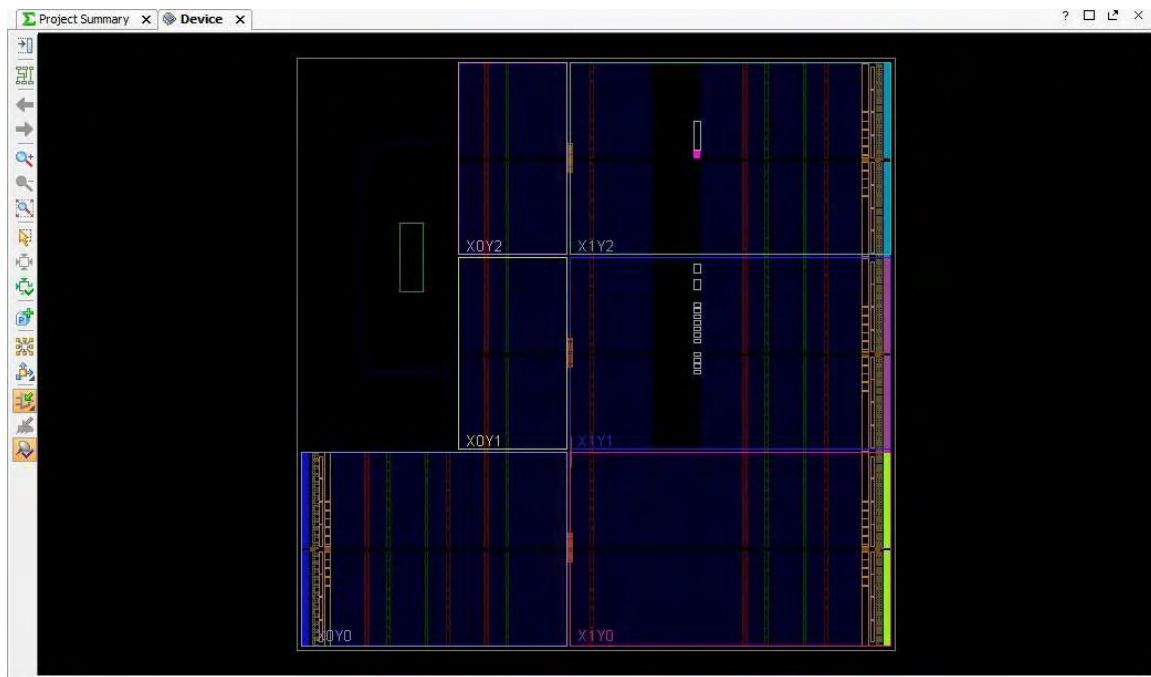
3.2.4 Επεξεργασία του αρχείου περιορισμού

Αρχικά ανοίγουμε την σχεδίαση που συνθέσαμε. Για να γίνει αυτό πατάμε στο κουμπί “Open Synthesized design” του Flow Navigator. Το κουμπί αυτό φαίνεται στο σχήμα 33.



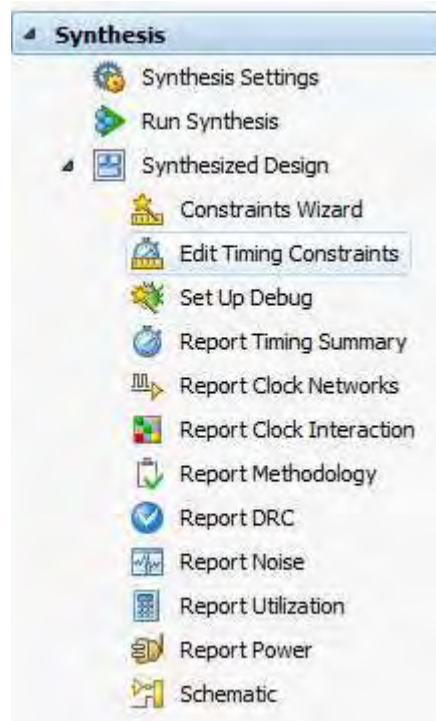
Σχήμα 33. Άνοιγμα της σχεδίασης που έχει συνθεθεί.

Μόλις ανοίξει η σχεδίαση, θα δούμε ένα παράθυρο που έχει την μορφή που φαίνεται στο σχήμα 34.



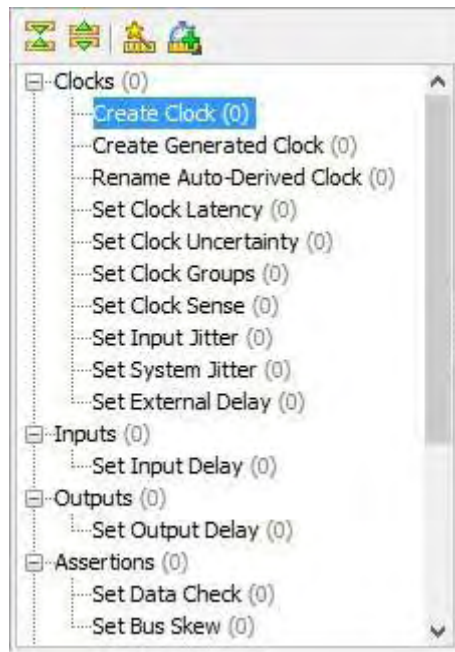
Σχήμα 34. Κάτοψη του επιλεγμένου FPGA.

Έτσι είμαστε σίγουροι ότι η σχεδίαση άνοιξε. Στην συνέχεια, από το Flow Navigator πατάμε το κουμπί “Edit Timing constraints”, ώστε να προσθέσουμε την περίοδο του ρολογιού. Το κουμπί αυτό φαίνεται στο σχήμα 35.



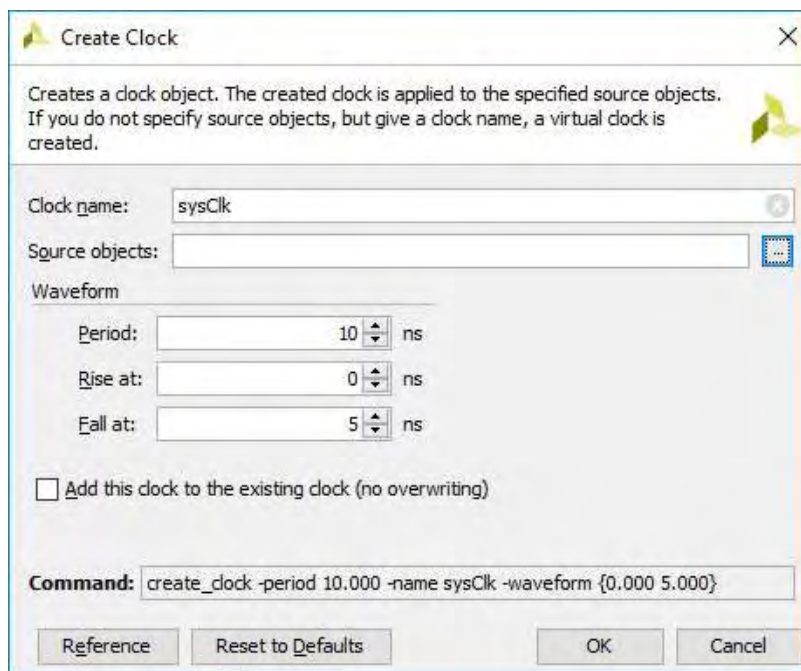
Σχήμα 35. Κουμπί για την επεξεργασία του αρχείου περιορισμών χρονισμού της σχεδίασης.

Το εργαλείο θα αλλάξει παράθυρο, όπου το νέο παράθυρο θα λέγεται “Timing constraints”. Για την δημιουργία του περιορισμού πατάμε διπλό click στην επιλογή “Create Clock”. Η επιλογή αυτή φαίνεται στο σχήμα 36.



Σχήμα 36. Επιλογή για τον ορισμό της συχνότητας του ρολογιού.

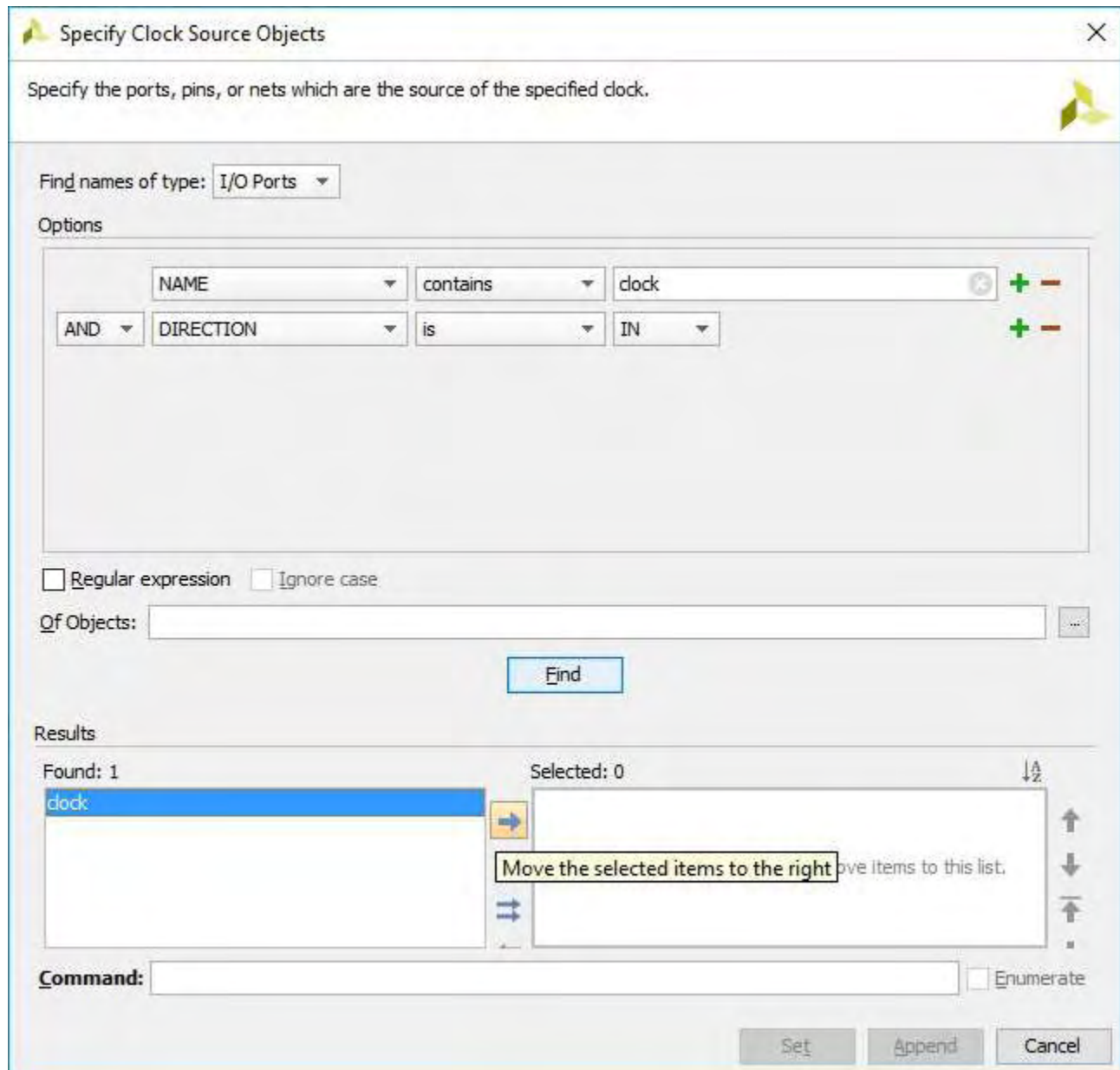
Έτσι θα ανοίξει το “Create Clock” wizard. Για όνομα του ρολογιού βάζουμε “sysClk» και μετά πατάμε στο κουμπί με τις τρεις τελείες που υπάρχει στα δεξιά του πεδίου ”Source Objects”. Το wizard και οι αλλαγές που κάναμε φαίνονται στο σχήμα 37.



Σχήμα 37. Παράθυρο εισαγωγής συχνότητας του ρολογιού και αντιστοίχισης με το ρολόι της σχεδίασης.

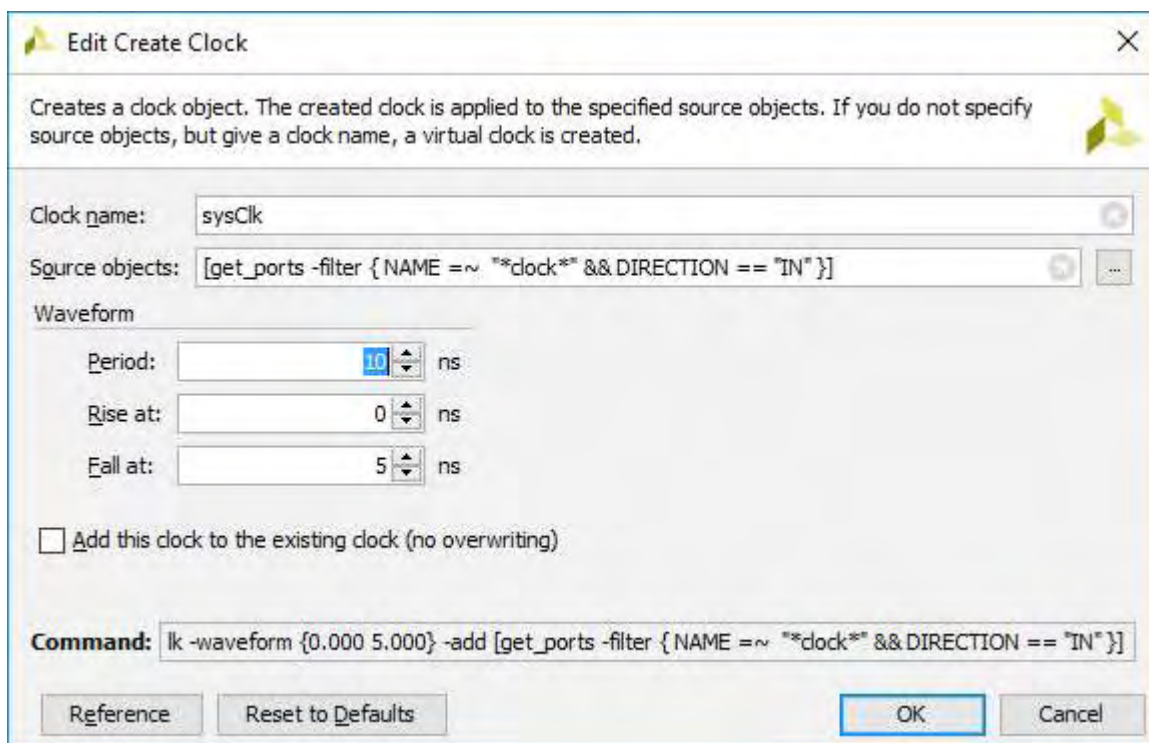
Όταν πατήσουμε το κουμπί με τις τρεις τελείες θα ανοίξει το παράθυρο με τίτλο “Specify Clock Source Objects”. Στο πρώτο πεδίο του παραθύρου γράφουμε “clock” και πατάμε το κουμπί “Find”, για να βρούμε το ρολόι της σχεδίασης που γράψαμε. Όταν φανερωθεί το σήμα του ρολογιού στο πλαίσιο “Found” στο κάτω μέρος του παραθύρου, τότε το επιλέγουμε και μετά πατάμε στο κουμπί που έχει ένα αριστερό βέλος. Μετά

πατάμε το κουμπί “OK” και ο περιορισμός συνδέεται με το ρολόι της σχεδίασης. Το τελευταίο παράθυρο φαίνεται στο σχήμα 38.



Σχήμα 38. Αναζήτηση του ρολογιού της σχεδίασης.

Στο προηγούμενο παράθυρο, που παραμένει ανοιχτό, ορίζουμε την περίοδο του ρολογιού και μετά πατάμε το κουμπί “OK”. Έτσι ολοκληρώνεται και η εισαγωγή της περιόδου του ρολογιού, άρα και η διαδικασία εισαγωγής του περιορισμού. Η αλλαγές της περιόδου, φαίνονται στο σχήμα 39.



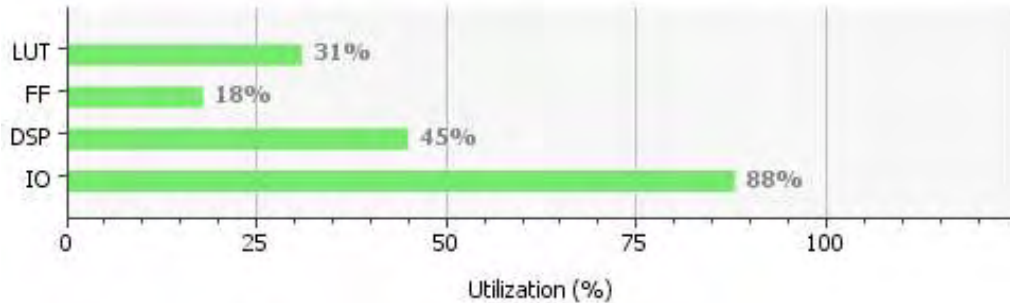
Σχήμα 39. Ορισμός της περιόδου του ρολογιού στα 10 ns και της συχνότητας στα 100 MHz.

Όπως φαίνεται και στο σχήμα, διατηρούνται οι τιμές των πεδίων. Έτσι ορίζεται η περίοδος του ρολογιού στα 10 ns, δηλαδή στα 100 MHz. Πατάμε το κουμπί “Apply” στο κάτω μέρος του παραθύρου “Timing constraints” και ο περιορισμός εγκαθιστάτε στην σχεδίαση. Στην συνέχεια ακολουθούνται ξανά τα βήματα που επιγράφηκαν στην ενότητα 3.2.2 για να επανασυνθεθεί η σχεδίαση λαμβάνοντας υπόψιν και τους περιορισμούς του ρολογιού. Στην επόμενη ενότητα (ενότητα 3.3), όπου παρουσιάζουμε τις αναφορές της σχεδίασης, θα δούμε ότι ο περιορισμός του ρολογιού τηρείται.

3.3 Αναφορές σχεδίασης

Κατά την ολοκλήρωση της σύνθεσης της σχεδίασης, παράχθηκαν αναφορές που ορίζουν το πλήθος των καταναλισκόμενων πόρων του επιλεγμένου FPGA. Από τις αναφορές αυτές διαπιστώθηκε ότι η σχεδίαση μπορεί να υποστηριχθεί από το επιλεγμένο FPGA. Στο σχήμα 40 φαίνεται η συνολική κατανάλωση πόρων, ενώ στο σχήμα 41 φαίνεται αναλυτικά και ανά υποκύκλωμα οι πόροι που καταναλώνονται.

Resource	Utilization	Available	Utilization %
LUT	16691	53200	31.37
FF	19261	106400	18.10
DSP	100	220	45.45
IO	175	200	87.50

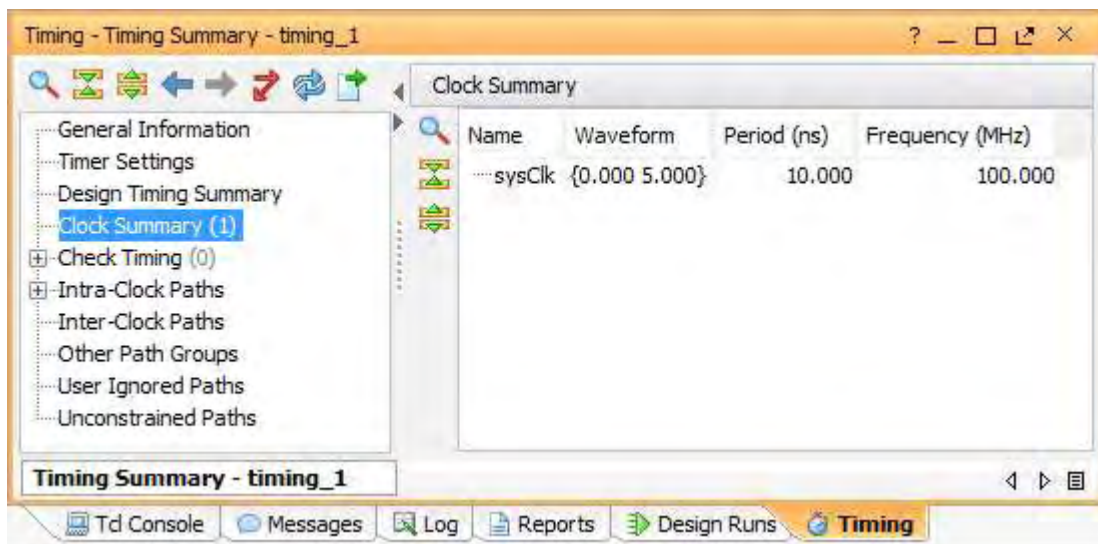


Σχήμα 40. Δεσμευόμενοι πόροι σε ποσοτική και ποσοστιαία μορφή.

Name	Slice LUTs (53200)	Slice Registers (106400)	F7 Muxes (26600)	F8 Muxes (13300)	DSPs (220)	Bonded IOB (200)	BUFGCTRL (32)
phy_layer_top_level	16691	19261	328	96	100	175	1
symbol_former_component (symbol_former)	1367	3475	89	0	0	0	0
pilot_component (pilot)	83	36	1	0	0	0	0
signal_field_former_component (signal_field_former)	22	28	0	0	0	0	0
scrambler_component (scrambler)	1	9	0	0	0	0	0
puncture_component (puncture)	7	7	0	0	0	0	0
preamble_component (preamble)	129	13	3	0	0	0	0
interleaver_component (interleaver)	1855	458	223	96	0	0	0
rom_96 (rom_96_addresses)	0	2	0	0	0	0	0
rom_48 (rom_48_addresses)	18	10	0	0	0	0	0
rom_288 (rom_288_addresses)	18	18	0	0	0	0	0
rom_192 (rom_192_addresses)	4	2	0	0	0	0	0
ram_288 (ddr_ram_288)	1687	300	222	96	0	0	0
ifft_component (ifft)	12560	12701	13	0	100	0	0
fsm_component (fsm)	231	92	0	0	0	0	0
data_mapper_top_level_component (data_mapper_top_level)	436	67	0	0	0	0	0
data_mapper_B (data_mapper_53)	230	33	0	0	0	0	0
data_mapper_A (data_mapper)	206	33	0	0	0	0	0
data_field_former_component (data_field_former)	55	21	0	0	0	0	0
ce_component (convolutional_encoder)	5	9	0	0	0	0	0
GI_and_windowing_component (GI_and_windowing)	15	2381	0	0	0	0	0

Σχήμα 41. Δεσμευόμενοι πόροι αναλυτικά για το κάθε κύκλωμα της σχεδίασης.

Μετά την δεύτερη σύνθεση, όπου λήφθηκαν υπόψιν και οι περιορισμοί του ρολογιού, η χρήση πόρων παρέμεινε ίδια. Επιπλέον επιβεβαιώθηκε ότι η σχεδίαση ικανοποιεί τους συγκεκριμένους περιορισμούς. Μερικά στιγμιότυπα από τις αναφορές του χρονισμού της σχεδίασης φαίνονται στα σχήματα 42 και 43. Στο πρώτο φαίνεται η περίοδος και συχνότητα του ρολογιού. Δηλαδή φαίνεται ότι το ρολόι έχει περίοδο 10 ns και συχνότητα 100 MHz. Ενώ στο δεύτερο φαίνεται ότι τηρούνται οι περιορισμοί του ρολογιού και μερικές λεπτομέρειες χρονισμού.



Σχήμα 42. Χαρακτηριστικά του ρολογιού της σχεδίασης.



Σχήμα 43. Τήρηση των περιορισμών του ρολογιού.

4 Υλοποίηση του φυσικού επιπέδου του πομπού του πρωτοκόλλου IEEE 802.11a

Εκτός από τα συστήματα που ορίζει το πρωτόκολλο, υλοποιούμε και μερικά επιπλέον συστήματα. Τα συστήματα αυτά είναι η μηχανή πεπερασμένων καταστάσεων (Finite state machine (FSM)) και τα συστήματα που σειριοποιούν τα δεδομένα των πεδίων preamble, signal και data. Με την χρήση αυτών των επιπρόσθετων κυκλωμάτων επιτυγχάνεται η ζητούμενη ροή δεδομένων. Οι σχεδιάσεις όλων των συστημάτων γράφονται σε γλώσσα VHDL και περιγράφονται σε αυτό το κεφάλαιο. Όμως για την κατανόηση των σχεδιάσεων πρέπει πρώτα να έχει μελετηθεί η ενότητα 2.1, όπου περιγράφονται οι ζητούμενες λειτουργίες των συστημάτων αυτών.

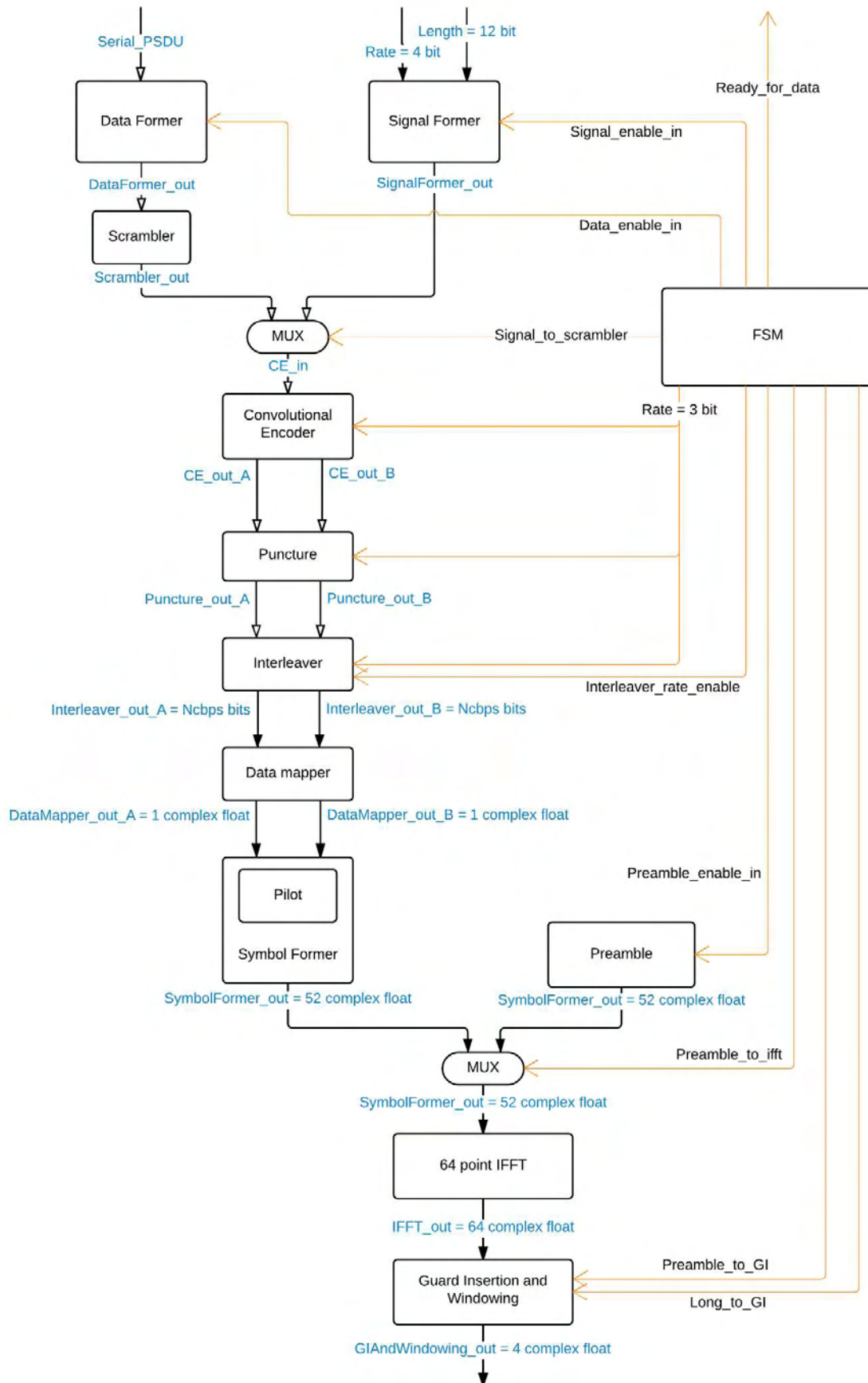
4.1 Ανώτερο επίπεδο σχεδίασης του επιπέδου PLCP

Το ανώτερο ιεραρχικό επίπεδο διασυνδέει όλα τα υλοποιημένα κυκλώματα μαζί. Η διασύνδεση γίνεται με τέτοιο τρόπο που καθιστά δυνατή την ζητούμενη ροή δεδομένων. Η ροή δεδομένων χειρίζεται από την FSM του συστήματος, η οποία περιγράφεται στην επόμενη ενότητα (ενότητα 4.2). Εκτός της απλής διασύνδεσης στην ευθύ ροή δεδομένων, όπου τα κυκλώματα συνδέονται με την σειρά. Στο ανώτερο επίπεδο υλοποιούμε και δύο πολυπλέκτες. Ο ένας από αυτούς τοποθετείται πριν την είσοδο του συνελκτικού κωδικοποιητή, ώστε να

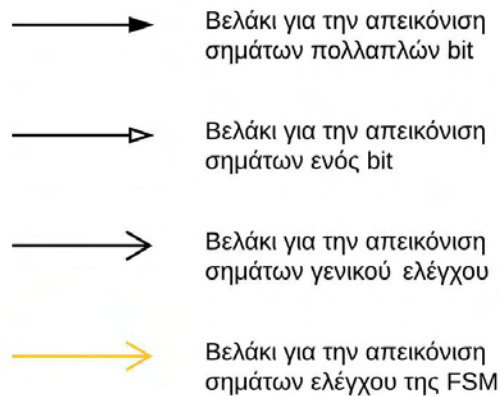
επιλέγεται η κατάλληλη. Οι δυνατές εισόδους του συνελκτικού κωδικοποιητή προέρχονται από την έξοδο του κυκλώματος συνένωσης του πεδίου signal ή από του scrambler. Από την άλλη πλευρά, ο δεύτερος πολυπλέκτης τοποθετείται πριν την είσοδο του κυκλώματος του αντίστροφου γρήγορου μετασχηματισμού Fourier. Ο πολυπλέκτης αυτός στέλνει στο κύκλωμα μετασχηματισμού την έξοδο του κυκλώματος του preamble ή του symbol former.

Η είσοδος του ανώτερου επιπέδου προέρχεται απευθείας από το επίπεδο MAC. Συγκεκριμένα, λαμβάνει τα πεδία rate, PSDU_length και PSDU όσο αναφορά τα δεδομένα. Το PSDU το λαμβάνει σειριακά, ένα bit σε κάθε κύκλο ρολογιού. Επιπρόσθετα, λαμβάνει το σήμα pad_bits_length και το σήμα εγκυρότητας της εισόδου του PSDU. Από την άλλη πλευρά η έξοδος του αποτελείται από 4 φανταστικούς αριθμούς, οι οποίοι προέρχονται από το κύκλωμα εισαγωγής τιμών φύλαξης και παραθύρωσης. Οι τέσσερις αυτές τιμές διαθέτουν πραγματικό και φανταστικό μέρος, όπου το κάθε ένα από αυτά έχει μέγεθος 18 bit. Οπότε συνολικά τα δεδομένα εξόδου θα απαρτίζονται από 144 bits. Τα δεδομένα εξόδου θα συνοδεύονται από το αντίστοιχο σήμα εγκυρότητας. Έκτος όμως από τα σήμα εξόδου που αναφέραμε, το ανώτερο επίπεδο εξάγει ένα ακόμα, το οποίο προέρχεται από την FSM και λέγεται ready_for_data. Το σήμα αυτό στέλνεται στο προηγούμενο επίπεδο του συστήματος επικοινωνίας και το ενημερώνει ότι το παρών επίπεδο είναι έτοιμο να λάβει το πεδίο PSDU. Το σχηματικό διάγραμμα του ανώτερου επιπέδου φαίνεται στο σχήμα 44. Το κάθε ένα από τα κυκλώματα που υλοποιήθηκαν θα περιγραφεί στην συνέχεια αυτού του κεφαλαίου.

Τα βελάκια με άσπρες κεφαλές υποδεικνύουν ότι το σήμα που αναπαριστούν έχει μέγεθος ενός bit. Ενώ τα βελάκια με μαύρες κεφαλές υποδεικνύουν ότι το σήμα έχει μέγεθος μεγαλύτερο του ενός bit και σε κάθε περίπτωση το μέγεθός του θα σημειώνεται στο σχήμα. Επιπλέον, κάθε καταχωρητής λαμβάνει το σήμα του ρολογιού, του reset και της ενεργοποίησης, τα οποία παραλείπονται για χάρη απλότητας του σχήματος. Σε ορισμένες περιπτώσεις η μετατροπή των σημάτων ελέγχου θα σημειώνεται στο κείμενο και στα σχήματα για κάθε έναν καταχωρητή. Τα παραλλαγμένα σήματα ελέγχου αλλά και όσα από τα βασικά σήματα ελέγχου υπάρχουν στα σχήματα, θα απεικονίζονται με βελάκια που διαθέτουν ανοιχτή κεφαλή. Συγκεκριμένα τα σήματα ελέγχου της FSM φαίνονται με ανοιχτές κεφαλές και πορτοκαλί χρώμα. Σύμφωνα με τα παραπάνω θα απεικονίζονται όλα τα σχηματικά διαγράμματα που θα δούμε σε αυτό το κεφάλαιο. Οι κατηγορίες των απεικονιζόμενων σημάτων σε αντιστοιχία με τα βελάκια φαίνεται στο σχήμα 45.



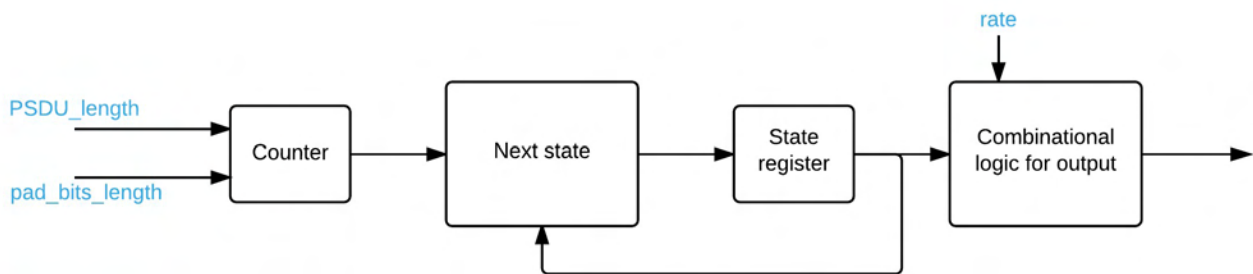
Σχήμα 44. Σχηματικό διάγραμμα του ανώτατου ιεραρχικού επιπέδου.



Σχήμα 45. Βελάκια που χρησιμοποιούνται σε όλα τα σχηματικά διαγράμματα της εργασίας.

4.2 Μηχανή πεπερασμένων καταστάσεων

Οι χρόνοι που χρειάζεται το κάθε στάδιο είναι προκαθορισμένοι. Για την επίτευξη της ζητούμενης ροής δεδομένων χρησιμοποιείται μία FSM, η οποία ελέγχεται από έναν μετρητή. Αυτός ο μετρητής βοηθάει στην εναλλαγή μεταξύ των καταστάσεων. Το σύνολο των καταστάσεων είναι 16. Όταν ολοκληρωθεί η αποστολή ενός ολόκληρου PLCP πακέτου, η FSM επανεκκινεί το κύκλωμα. Έτσι επαναφέρεται στην αρχική του κατάσταση και μπορεί να επεξεργαστεί και να παραγάγει το επόμενο πακέτο. Η FSM που υλοποιήθηκε έχει την δομή μίας Moore μηχανής καταστάσεων και το σχηματικό της διάγραμμα φαίνεται στο σχήμα 46.



Σχήμα 46. Σχηματικό διάγραμμα μηχανής πεπερασμένων καταστάσεων.

4.3 Προοίμιο

Το preamble στην τελική του μορφή, όπου θα είναι έτοιμο για αποστολή, αποτελείται από πολλές τιμές. Όμως το πρωτόκολλο έχει ήδη ενσωματωμένα τα συστήματα που χρειάζεται ώστε το πρόθεμα να έρθει σε μορφή κατάλληλη για αποστολή. Για αυτό τον λόγο έχει ορίσει τις τιμές του προθέματος σε πιο πρώιμο στάδιο επεξεργασίας. Με αυτόν τον τρόπο αποφεύγεται η χρήση επιπρόσθετης μνήμης για την αποθήκευση του προθέματος στην τελική του μορφή, το οποίο έχει σχεδόν τριπλάσιο μέγεθος από την αρχική του κατάσταση. Αυτό κατά κύριο λόγο συμβαίνει γιατί στο preamble, όπως και στα άλλα δύο πεδία, προστίθεται ένα κυκλικό πρόθεμα. Συγκεκριμένα, τα συστήματα που χρειάζεται το preamble ώστε να έρθει σε μορφή κατάλληλη για

αποστολή είναι το σύστημα του αντίστροφου γρήγορου μετασχηματισμού Fourier και το σύστημα του εισαγωγής τιμών φύλαξης και παραθύρωσης, όπου περιγράφονται στις ενότητες 4.13 και 4.14, αντίστοιχα. Με την βοήθεια του αντίστροφου γρήγορου μετασχηματισμού Fourier θα μεταφερθούν στο πεδίο του χρόνου, και στην συνέχεια με τη βοήθεια του συστήματος εισαγωγής τιμών φύλαξης και παραθύρωσης θα τους προσαρτηθεί ένα κυκλικό πρόθεμα και θα επεκταθεί το πλήθος των τιμών που μεταφέρουν.

Όπως οι διαμορφωμένες τιμές των δεδομένων έτσι και εδώ, οι τιμές είναι μιγαδικές, μιας και οι δύο βρίσκονται στη ζώνη συχνοτήτων που απαιτεί το κανάλι επικοινωνίας. Παρατηρείται από τις σχέσεις 1 και 2 της ενότητας 2.1.1 ότι οποιαδήποτε τιμή του προθέματος είναι είτε 1 είτε -1. Όμως οι τιμές του πρώτου τμήματος θα πολλαπλασιαστούν με έναν παράγοντα κανονικοποίησης. Το αποτέλεσμα του πολλαπλασιασμού αυτού έχει προϋπολογιστεί. Οπότε, συμπεραίνεται ότι υπάρχουν τέσσερις διαφορετικές τιμές, από τις οποίες αποτελούνται τα σύμβολα των δύο τμημάτων. Οι τιμές αυτές αποθηκεύονται σε τέσσερις καταχωρητές, ώστε να είναι διαθέσιμες όποτε χρειαστούν.

Όταν ενεργοποιηθεί το σήμα εκκίνησης, όπου το κύκλωμα του preamble το λαμβάνει ως είσοδο, το σύμβολο του πρώτου τμήματος του προθέματος θα σταλθεί απευθείας στην είσοδο του συστήματος του αντίστροφου γρήγορου μετασχηματισμού Fourier. Για την αποστολή του συμβόλου του δεύτερου τμήματος θα χρειαστεί να περιμένουμε λίγο, ώστε το πρώτο σύμβολο να μπορέσει να επεξεργαστεί πλήρως από το σύστημα εισαγωγής τιμών φύλαξης και παραθύρωσης. Η αναμονή θα διαρκέσει 39 κύκλους ρολογιού και θα μετριέται με την βοήθεια του μετρητή που βρίσκεται στο κύκλωμα της FSM. Όταν περάσει το χρονικό διάστημα που απαιτείται, αποστέλλεται και το σύμβολο του δεύτερου τμήματος στο σύστημα του αντίστροφου γρήγορου μετασχηματισμού Fourier. Η έξοδος του συστήματος θα συνοδεύεται και από το αντίστοιχο σήμα εγκυρότητας, όπου και αυτό ελέγχεται με την βοήθεια του μετρητή. Όταν ολοκληρωθεί η επεξεργασία ολόκληρου του προθέματος του πακέτου, τότε θα αρχίσει και η επεξεργασία των πεδίων signal και data.

Πρέπει να σημειωθεί ότι το σύστημα symbol former μοιάζει με το σύστημα παραγωγής του προθέματος του πακέτου, αλλά δεν χρησιμοποιήθηκε για την ενοποίηση των τιμών των συμβόλων του, γιατί έχει ενσωματωμένη την λειτουργία εισαγωγής των πιλοτικών τιμών, οι οποίες δεν θα χρειαστούν να τοποθετηθούν μέσα στο πρόθεμα. Επιπλέον, η διαφοροποίηση του symbol former και του κυκλώματος του preamble βοηθούν στην απλοποίηση της συνολικής σχεδίασης του φυσικού επιπέδου.

4.4 Κύκλωμα συνένωσης του πεδίου signal

Το σύστημα αυτό πρέπει να μετατρέψει τα πεδία που λαμβάνει παράλληλα, σε μία έννοια σειριακή ροή δεδομένων. Τα πεδία που απαρτίζουν το πεδίο Signal είναι τα rate, reserved, PSDU_length, parity και tail. Τα πεδία αυτά περιγράφονται στην ενότητα 2.1.2. Η δομή του πεδίου signal που ορίζει το πρωτόκολλο φαίνεται στον πίνακα 7.

Πίνακας 7. Πεδίο Signal.

Πεδίο rate 4 bits	Πεδίο reserved 1 bit	Πεδίο PSDU_length 12 bits	Πεδίο parity 1 bit	Πεδίο tail 6 bits
----------------------	-------------------------	------------------------------	-----------------------	----------------------

Τα πεδία αυτά προέρχονται παράλληλα από το ανώτερο επίπεδο του συστήματος επικοινωνίας, το επίπεδο MAC. Για την σειριακή εξαγωγή τους θα χρειαστεί να χρησιμοποιηθούν μερικοί buffer. Οι buffer υλοποιούνται ως καταχωρητές ολίσθησης, όπου ο καθένας έχει το μέγεθος που απαιτείται ώστε να καλύπτει το πεδίο που θα αποθηκεύει. Τα μεγέθη των καταχωρητών ολίσθησης για το κάθε ένα πεδίο φαίνονται στον πίνακα 7, όπου το κάθε bit θα αντιστοιχείται σε ένα στοιχείο της μνήμης του. Αφού αποθηκευτούν όλες οι τιμές των πεδίων στους αντίστοιχους buffer, θα χρειαστεί να εξαχθούν με την σειρά οι κατάλληλες τιμές. Η σειρά εξαγωγής είναι προκαθορισμένη από το πρωτόκολλο και είναι ίδια με αυτή του πίνακα 7.

Για να ολισθήσουμε κατάλληλα τους buffers και να εξαχθούν οι ζητούμενες τιμές, θα χρησιμοποιηθεί ένας μετρητής, ένας πολυπλέκτης και τα σήματα που υποδεικνύουν το πεδίο που εξάγεται κάθε χρονική στιγμή.

Τα τελευταία σήματα τα ονομάζουμε `rate_enable`, `reserved_enable`, `length_enable`, `parity_enable`, `tail_enable`, όπου το καθένα από αυτά αντιστοιχεί στο πεδίο που προδίδει το όνομά του. Όταν ένα από αυτά τα σήματα ενεργοποιηθεί, τότε ο κατάλληλος `buffer` ολισθαίνει και ο μετρητής αυξάνεται μέχρι την τερματική τιμή που ορίζει ο πολυπλέκτης. Οι τερματικές τιμές μέχρι τις οποίες πρέπει να μετρήσει ο μετρητής είναι ίσες με το μέγεθος του κάθε πεδίου.

Συγκεκριμένα, η ροή δεδομένων μέσα στο κύκλωμα του συστήματος συνένωσης του πεδίου `Signal` είναι η εξής. Όταν ενεργοποιηθεί το σήμα εκκίνησης του συστήματος που ονομάζεται `enable_in`, ενεργοποιείται και το σήμα `rate_enable`. Έτσι σε κάθε επόμενο κύκλο ρολογιού, μέχρις ότου ο μετρητής φτάσει να είναι ίσος με 3, ο `buffer` του πεδίου `rate` ολισθαίνει. Στην έξοδο του συστήματος στέλνεται η τιμή που διαθέτει ο `buffer` και ταυτόχρονα ενεργοποιείται και το σήμα εγκυρότητας της εξόδου. Όταν τελικά ο μετρητής φτάσει να είναι ίσος με 3, το σήμα `rate_enable` και ο ίδιος μηδενίζεται, ενώ ενεργοποιείται το σήμα `reserved_enable`. Το πεδίο `reserved` έχει μέγεθος ενός `bit` και η τιμή του είναι μηδενική, έτσι δεν χρησιμοποιείται κάποιος `buffer`. Στον επόμενο κύκλο ρολογιού πριν ο μετρητής προλάβει να αυξήσει την τιμή του ξανά μηδενίζεται και μαζί με αυτόν μηδενίζεται και το σήμα `reserved_enable`, ενώ ενεργοποιείται το σήμα `length_enable`. Για 12 κύκλους ρολογιού ο `buffer` του πεδίου `length` ολισθαίνει και στέλνει την τιμή του στην έξοδο του συστήματος, ταυτόχρονα αυξάνεται και ο μετρητής. Μόλις ο μετρητής φτάσει να είναι ίσος με 11, το σήμα `length_enable` και ο ίδιος μηδενίζονται, ενώ ενεργοποιείται το σήμα `parity_enable`. Όπως και το πεδίο `reserved` έτσι και το `parity` έχει μέγεθος ενός `bit`. Έτσι η τιμή του `parity` θα σταλθεί στην έξοδο του συστήματος απευθείας. Στον επόμενο κύκλο ρολογιού ο μετρητής και το σήμα `parit_enable` μηδενίζονται, ενώ ενεργοποιείται και το τελευταίο σήμα που είναι το `tail_enable`. Η τιμή του είναι μηδενική, άρα δεν θα χρειαστεί να αποθηκευτεί σε κάποιον `buffer`. Οπότε μέχρι ο μετρητής να μετρήσει 6 κύκλους ρολογιού, στην έξοδο του συστήματος θα στέλνεται ένα μηδενικό `bit`. Όταν τελικά ο μετρητής φτάσει να είναι ίσος με 5, μηδενίζεται. Μαζί με αυτόν μηδενίζεται το σήμα `tail_enable` και το σήμα εγκυρότητας εξόδου. Με τον τρόπο αυτόν ολοκληρώνεται η σειριακή εξαγωγή του πεδίου `signal`.

Όπως θα φανεί στην επόμενη ενότητα, μέχρι στιγμής χρησιμοποιείται η ίδια δομή με αυτή του συστήματος συνένωσης του πεδίου δεδομένων. Και αυτό γιατί χρειάζεται να υλοποιηθεί όμοια λειτουργία. Μόνο που στην περίπτωση του παρόντος συστήματος χρησιμοποιούνται `buffers` για την σειριακή εξαγωγή των τιμών των πεδίων, ενώ στο άλλο σύστημα δεν χρειάζεται να χρησιμοποιηθούν `buffers` μιας και το πεδίο `PSDU` εισάγεται σειριακά και τα υπόλοιπα πεδία έχουν εξ' αρχής μηδενικές τιμές. Επιπλέον, στο σύστημα συνένωσης του πεδίου `signal` πρέπει να υπολογιστεί και η τιμή του σήματος `parity`. Το σήμα αυτό είναι άσσος όταν το πλήθος των άσων μέσα στα πεδία `rate` και `PSDU_length` είναι μονός αριθμός, ενώ έχει μηδενική τιμή όταν το πλήθος των άσων στα ίδια πεδία είναι ζυγός αριθμός.

Συνολικά χρησιμοποιήθηκαν 2 καταχωρητές ολίσθησης για τα πεδία `rate` και `PSDU_length`. Ένας καταχωρητής για την αποθήκευση της τιμής του πεδίου `parity`, ο οποίος στην είσοδό του λαμβάνει το αποτέλεσμα της λογικής πράξης XOR μεταξύ της τιμής του και της τιμής που δίνει ο `buffer` του πεδίου `rate` ή του πεδίου `PSDU_length`. Ένας μετρητής και ένας πολυπλέκτης για την διαχείριση της ροής των δεδομένων εντός του συστήματος αυτού. Τα κατάλληλα σήματα για τον χειρισμό του πολυπλέκτη, που υποδεικνύουν το πιο πεδίο επεξεργάζεται την κάθε χρονική στιγμή. Τα σήματα αυτά αποθηκεύονται σε 4 διαφορετικούς καταχωρητές, οι οποίοι διαθέτουν σήμα ενεργοποίησης και σύγχρονο σήμα επανεκκίνησης. Τα σήματα ελέγχου των καταχωρητών αυτών χειρίζονται από τον μετρητή. Ενώ χρησιμοποιήθηκε ένας ακόμα καταχωρητής για την αποθήκευση της τιμής του σήματος εγκυρότητας της εξόδου, η οποία αν τουλάχιστον ένα από τα σήματα ενεργοποίησης των πεδίων είναι άσσος, τότε και αυτή γίνεται άσσος. Τέλος, χρησιμοποιήθηκε και ένας καταχωρητής για την αποθήκευση της εξόδου του συστήματος, ο οποίος διαθέτει την τιμή των `buffer` όταν τα κατάλληλα πεδία είναι ενεργά, ενώ διαθέτει μηδενική τιμή σε κάθε άλλη περίπτωση. Όπως φαίνεται χρησιμοποιήθηκαν 2 καταχωρητές ολίσθησης, 7 κανονικοί καταχωρητές, ένας μετρητής και ένα πολυπλέκτης για να επιτευχθεί η ζητούμενη λειτουργία του συστήματος συνένωσης του πεδίου `signal`.

4.5 Κύκλωμα συνένωσης του πεδίου data

Ο σκοπός του συστήματος συνένωσης του πεδίου data είναι όμοιος με του αντίστοιχου συστήματος για την συνένωση του πεδίου signal. Συγκεκριμένα, ο στόχος του είναι η συνένωση των πεδίων service, PSDU, tail και pad, ώστε να σχηματίσουν μία ενιαία ροή δεδομένων (bit stream). Το σύνολο των παραπάνω πεδίων ονομάζεται πεδίο data ή PPDU. Τα 4 αυτά πεδία και η χρησιμότητά τους περιγράφονται στην ενότητα 2.1.3. Η δομή του πεδίου data που ορίζει το πρωτόκολλο φαίνεται στον πίνακα 8.

Πίνακας 8. Πεδίο Data.

Πεδίο service 16 bits	Πεδίο PSDU	Πεδίο tail 6 bits	Πεδίο pad
--------------------------	------------	----------------------	-----------

Τα μεγέθη των πεδίων PSDU και pad είναι μεταβλητά και καθορίζονται από ειδικά σήματα εισόδου, τα οποία θα αναφερθούν στην συνέχεια. Για την σωστή ροή των δεδομένων χρησιμοποιείται ένας μετρητής. Αυτός ο μετρητής αυξάνεται μέχρι το σημείο που απαιτεί το κάθε ένα πεδίο και όταν το φτάσει μηδενίζεται και ξεκινάει από την αρχή. Ο μετρητής αυξάνεται κάθε φορά που ένα από τα πεδία είναι ενεργό και πρέπει να εξαχθεί. Πιο συγκεκριμένα, ο μετρητής θα πρέπει να μετρήσει 16 κύκλους ρολογιού για το πεδίο service, όσο το μήκος του πεδίου PSDU που καθορίζεται από το σήμα PSDU_length του πεδίου signal, 6 κύκλους ρολογιού για το πεδίο tail και όσο το μήκος του πεδίου pad που καθορίζεται από το σήμα pad_bits_length. Η καταμέτρηση θα γίνει όσο αναφέραμε και με την ίδια σειρά. Τα σήματα PSDU_length και pad_bits_length προέρχονται από το ανώτερο επίπεδο του πρωτοκόλλου, δηλαδή από το MAC. Οι χρόνοι που θα χρειαστεί να μετρήσει ο μετρητής για κάθε ένα από τα πεδία φαίνονται στον πίνακα 9.

Πίνακας 9. Μήκος σε bits των πεδίων του Data.

Όνομα πεδίου	Κύκλοι ρολογιού = Μέγεθος πεδίου
Service	16
PSDU	PSDU_length * 8
Tail	6
Pad bits	pad_bits_length

Η ροή δεδομένων μέσα στο κύκλωμα του συστήματος συνένωσης του πεδίου Data είναι η εξής. Όταν ενεργοποιηθεί το σήμα εκκίνησης του συστήματός αυτού, το οποίο λαμβάνεται απευθείας από την FSM, ενεργοποιείται και το σήμα που σηματοδοτεί την εκκίνηση της εξαγωγής του πεδίου service. Το σήμα αυτό ονομάζεται service_enable και όταν ενεργοποιείται, επιτρέπεται η αύξηση του μετρητή, ενεργοποιείται το σήμα εγκυρότητας της εξόδου και εξάγεται η τιμή του πεδίου service. Το πεδίο service απαρτίζεται από 16 μηδενικά. Έτσι, για 16 κύκλους ρολογιού θα στέλνεται ένα μηδενικό bit στην έξοδο δεδομένων. Όταν στον τρέχον κύκλο ρολογιού η τιμή του μετρητή φτάσει να είναι ίση με 15, τότε στον επόμενο κύκλο ρολογιού ο μετρητής και το σήμα ενεργοποίησης του πεδίου service μηδενίζεται και ενεργοποιείται το σήμα data_enable. Το τελευταίο σήμα υποδεικνύει την εκκίνηση της εξαγωγής του πεδίου PSDU. Όσο το σήμα data_enable είναι ενεργό ο μετρητής αυξάνεται ξανά, το σήμα εγκυρότητας της εξόδου παραμένει ενεργό και το πεδίο PSDU εξάγεται σειριακά σε κάθε κύκλο ρολογιού. Όταν η τιμή του μετρητή γίνει ίση με το μήκος του πεδίου PSDU, δηλαδή ίση με την τιμή του σήματος PSDU_length, τότε το σήμα data_enable και ο μετρητής μηδενίζεται, ενώ το σήμα tail_enable ενεργοποιείται. Το σήμα tail_enable σηματοδοτεί την εκκίνηση εξαγωγής του πεδίου tail. Το πεδίο tail αποτελείται από 6 μηδενικά bit. Οπότε για 6 κύκλους ρολογιού ο μετρητής θα αυξάνεται, το σήμα εγκυρότητας της εξόδου θα παραμένει ενεργό και ένα μηδενικό bit θα στέλνεται στην έξοδο του συστήματος. Όταν ο μετρητής γίνει ίσος με 5, τότε το σήμα tail_enable και ο ίδιος μηδενίζεται, ενώ ενεργοποιείται το σήμα

pad_bits_enable. Το σήμα pad_bits_enable υποδεικνύει την εκκίνηση της εξαγωγή του πεδίου pad. Όπως το πεδίο PSDU έτσι και το πεδίο pad έχει μεταβλητό μέγεθος, το οποίο καθορίζεται από το σήμα pad_bits_length. Επομένως όσο ο μετρητής δεν έχει φτάσει την τιμή του pad_bits_length, το σήμα εγκυρότητας της εξόδου παραμένει ενεργό και σε κάθε ένα κύκλο ρολογιού εξάγεται ένα ακόμα μηδενικό bit. Όταν τελικά ο μετρητής γίνει ίσος με το σήμα pad_bits_length, τότε αυτός, το σήμα εγκυρότητας της εξόδου και το σήμα pad_bits_enable μηδενίζονται. Με αυτόν τον τρόπο ολοκληρώνεται η εξαγωγή του πεδίου data.

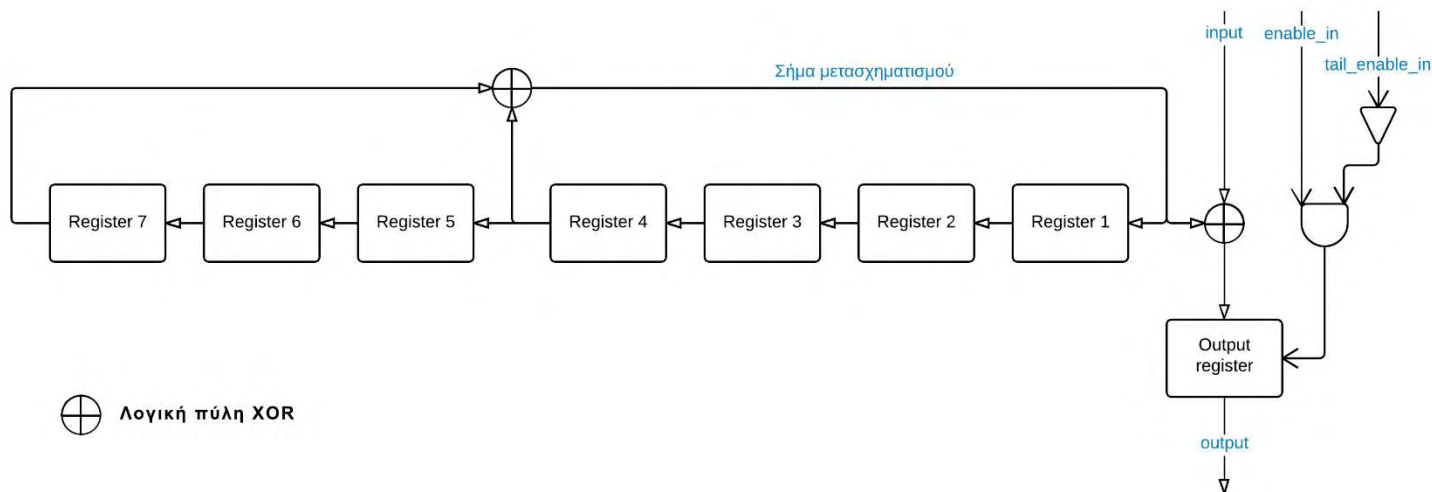
Όπως αναφέρθηκε, για την επίτευξη της παραπάνω ροής χρησιμοποιήθηκε ένας μετρητής. Όμως αυτός ο μετρητής πρέπει να φτάνει σε διαφορετικές τερματικές τιμές πριν επανεκκινηθεί. Για την επιλογή της κατάλληλης τερματικής τιμής χρησιμοποιείται και ένας πολυπλέκτης, οποίος επιλέγει την κατάλληλη βάση των σημάτων ενεργοποίησης του κάθε πεδίου. Συνολικά τα σήματα ενεργοποίησης που χρησιμοποιήθηκαν είναι τα service_enable, PSDU_enable, tail_enable και pad_bits_enable. Το σήμα του πεδίου PSDU και το αντίστοιχο σήμα εγκυρότητας, το PSDU_length και το pad_bits_length εισάγεται απευθείας από το ανώτερο επίπεδο του συστήματος επικοινωνίας. Συγκεκριμένα το πεδίο PSDU εισάγεται σειριακά στο κύκλωμα, ενώ τα υπόλοιπα διαβάζονται παράλληλα μόνο μία φορά, κατά την εκκίνηση της λειτουργίας του. Η τιμές της εξόδου αποθηκεύονται σε κατάλληλο καταχωρητή, ο οποίος εξάγει την τιμή του εισερχόμενου bit του πεδίου PSDU όταν το σήμα PSDU_enable είναι ενεργοποιημένο ή εξάγει ένα μηδενικό bit σε κάθε άλλη περίπτωση. Με την βοήθεια του σήματος εγκυρότητας της εξόδου το σήμα εξόδου παράγεται σωστά. Είναι σημαντικό να σημειωθεί ότι εξάγεται ένα ακόμα σήμα το οποίο υποδεικνύει ότι το τρέχον bit εξόδου αποτελεί μέρος του πεδίου tail, το οποίο ονομάζεται tail_enable_out. Το σήμα αυτό στέλνεται στο κύκλωμα του scrambler ώστε να μηδενίσει την τιμή του πεδίου tail μετά τον μετασχηματισμό της. Η χρησιμότητα αυτού του σήματος εξηγείται περισσότερο στην ενότητα 2.3.1, όπου περιγράφεται το σύστημα του scrambler.

4.6 Scrambler

Για την υλοποίηση της σχεδίασης του scrambler, αρχικά χρησιμοποιήθηκε ένας ανατροφοδοτούμενος καταχωρητής ολίσθησης (feedback shift register) μήκους 7 καταχωρητών, προκειμένου να αποθηκευτεί και να παραχθεί το σήμα μετασχηματισμού. Κατά την εκκίνηση, αυτός ο καταχωρητής ολίσθησης, τροφοδοτείται με τον μη μηδενικό σπόρο “1011101”. Ο συγκεκριμένος σπόρος δίνεται από το πρωτόκολλο και χρησιμοποιήθηκε ώστε να επαληθευτεί η σωστή παραγωγή του σήματος μετασχηματισμού. Κάθε χρονική στιγμή το σήμα μετασχηματισμού έχει την τιμή που προέρχεται από την λογική πράξη XOR ανάμεσα στις τιμές του τέταρτου και του έβδομου καταχωρητή του καταχωρητή ολίσθησης. Η τιμή αυτή ανατροφοδοτείται στον καταχωρητή ολίσθησης. Στον ίδιο κύκλο ρολογιού το σήμα μετασχηματισμού και το σήμα πληροφορίας περνάνε από μία ακόμα λογική πύλη XOR, το αποτέλεσμα της οποίας αποθηκεύεται απευθείας στον καταχωρητή εξόδου του κυκλώματος.

Εκτός όμως από την απομονωμένη υλοποίηση του scrambler, στο ίδιο κύκλωμα προσαρμόζουμε και τον μηδενισμό του πεδίου tail του PDDU. Η ενοποίηση των δύο κυκλωμάτων αφενός, βοηθάει στην εξοικονόμηση ενός κύκλου ρολογιού και αφετέρου, δεν αυξάνει το κρίσιμο μονοπάτι (critical path) μιας και έχουμε ορίσει ως κρίσιμο μονοπάτι αυτό ενός πολλαπλασιαστή και ενός προσθέτη στην σειρά. Η ενοποίηση των δύο κυκλωμάτων επιτυγχάνεται με την αλλαγή του σήματος ενεργοποίησης και επανεκκίνησης (reset) του καταχωρητή εξόδου του scrambler. Η αλλαγή γίνεται ως εξής, αντί η εγγραφή να γίνεται μόνο όταν το σήμα εγκυρότητας της εισόδου είναι ενεργό, η εγγραφή είναι δυνατή όταν το προηγούμενο σήμα είναι ενεργό και ταυτόχρονα, η είσοδος δεν προέρχεται από το πεδίο tail του data. Σε κάθε άλλη περίπτωση η έξοδος του scrambler θα είναι μηδενική, γιατί ο παραπάνω συνδυασμός σημάτων χρησιμοποιείται και ως σήμα επανεκκίνησης του καταχωρητή. Για την εξακρίβωση της παρουσίας του πεδίου tail ο scrambler λαμβάνει κατάλληλο σήμα από το κύκλωμα συνένωσης του πεδίου data, το οποίο ονομάζεται “tail_enable_in”. Η κατανόηση της αλλαγής αυτής γίνεται ευκολότερη με την βοήθεια του σχήματος 47. Το σήμα εγκυρότητας της εξόδου είναι ενεργό τον τρέχον κύκλο ρολογιού, μόνο αν στον προηγούμενο κύκλο ρολογιού το σήμα

εγκυρότητας εισόδου ήταν ενεργό. Το σχηματικό διάγραμμα του κυκλώματος που υλοποιήσαμε είναι αυτό του σχήματος 47.

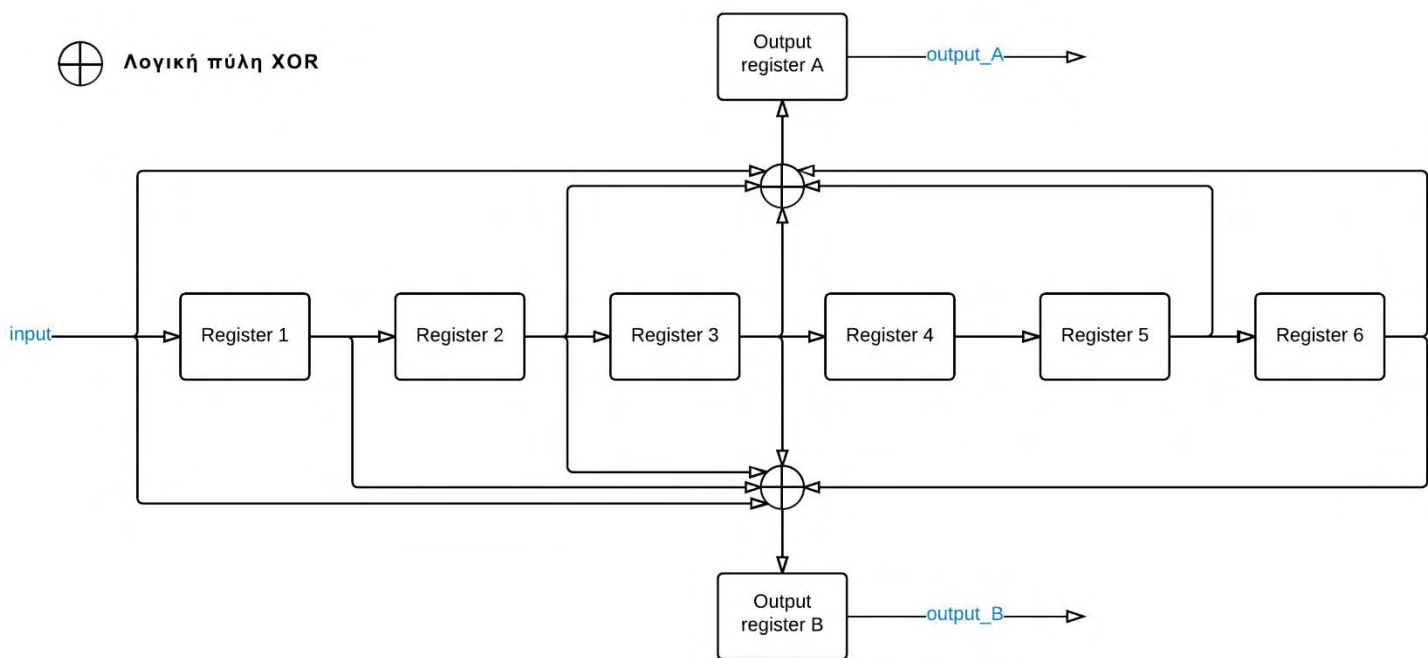


Σχήμα 47. Σχηματικό διάγραμμα του Scrambler.

4.7 ΣΥΝΕΛΙΚΤΙΚΟΣ ΚΩΔΙΚΟΠΟΙΗΤΗΣ

Τα πολυώνυμα γεννήτριες που χρησιμοποιεί ο συνελικτικός κωδικοποιητής για να παραγάγει τις εξόδους του συστήματος, χρειάζονται μόνο έναν καταχωρητή ολίσθησης για την υλοποίησή τους. Ο καταχωρητής ολίσθησης που θα χρησιμοποιηθεί, τροφοδοτείται με το σήμα εισόδου του συστήματος και έχει μήκος έξι καταχωρητών. Η είσοδος εισέρχεται σειριακά στο σύστημα, μιας και λαμβάνεται από την έξοδο του scrambler. Για να λάβουμε τα γεννήτρια πολυώνυμα που επιθυμούμε αρκεί μόνο να διαβαστούν οι τιμές συγκεκριμένων καταχωρητών. Για την ακρίβεια, για την δημιουργία της τιμής του πολυωνύμου S_A της σχέσης 4 από την ενότητα 2.3.2, το οποίο θα αποθηκευτεί στον καταχωρητή εξόδου του σήματος A, χρειάζεται να διαβαστεί και να περαστεί από μία λογική πύλη XOR η τιμή του δεύτερου, του τρίτου, του πέμπτου και του έκτου καταχωρητή αλλά και η τιμή της εισόδου. Αντίστοιχα, για την παραγωγή της τιμής του πολυωνύμου S_B της σχέσης 5 από την ίδια ενότητα, που θα αποθηκευτεί στον καταχωρητή της εξόδου B, πρέπει να διαβαστεί και να περαστεί από μία λογική πύλη XOR η τιμή του πρώτου, του δεύτερου, του τρίτου και του έκτου καταχωρητή αλλά και πάλι, η τιμή της εισόδου. Δηλαδή όπως ακριβώς φαίνεται και στο σχήμα 48.

Οι καταχωρητές εξόδου, A και B, θα διαθέτουν τις κωδικοποιημένες τιμές της τρέχον εισόδου του συστήματος, στον επόμενο κύκλο ρολογιού. Είναι σημαντικό να σημειωθεί ότι, για την επιβεβαίωση της εγκυρότητας των σημάτων έξοδου του συνελικτικού κωδικοποιητή χρησιμοποιείται μόνο ένα σήμα επιβεβαίωσης, το οποίο είναι η ολισθημένη τιμή, κατά έναν κύκλο ρολογιού, του σήματος εγκυρότητας εισόδου. Αυτό σημαίνει πως όταν την τρέχον χρονική στιγμή η είσοδος είναι έγκυρη, τότε στην επόμενη χρονική στιγμή και οι δύο εξοδοί θα είναι έγκυροι. Το σχηματικό του διάγραμμα του συνελικτικού κωδικοποιητή είναι αυτό του σχήματος 48.



Σχήμα 48. Σχηματικό διάγραμμα του συνελκτικού κωδικοποιητή.

4.8 Puncture

Στην υλοποίηση του puncture, αποθηκεύονται οι δύο τιμές του λαμβανόμενου κωδικοποιημένου σήματος σε δύο καταχωρητές εξόδου με ονόματα output A και output B. Η αποθήκευση στους καταχωρητές αυτούς γίνεται κάθε φορά που η είσοδος του συστήματος είναι έγκυρη. Έτσι αυτοί απλά θα ολισθαίνουν την έγκυρη είσοδο κατά ένα κύκλο ρολογιού. Προκειμένου να σχεδιάσουμε ένα κύκλωμα που παράγει την προσδοκώμενη έξοδο μόνο σε έναν κύκλο ρολογιού, η μεταβολή του ρυθμού μετάδοσης χειρίζεται πλήρως από τα σήματα εγκυρότητας της εξόδου. Είναι προφανές ότι αν χρειαστεί να διατηρηθεί ο ρυθμός κώδικα του συνελκτικού κωδικοποιητή, τότε στην έξοδο του κυκλώματος του puncture εκτός από την είσοδο, προωθούμε αυτούσιο και το σήμα εγκυρότητας της. Έτσι, το κωδικοποιημένο σήμα θα μεταφερθεί καθυστερημένο κατά έναν κύκλο ρολογιού, με ρυθμό κώδικα $1/2$, στον interleaver.

Στην περίπτωση που ο ρυθμός μετάδοσης χρειάζεται να είναι ίσος με $2/3$, χρησιμοποιείται ένας καταχωρητής, στον οποίο αποθηκεύεται η τιμή που προκύπτει από τη λογική πράξη NOT της τρέχον τιμής του. Ο καταχωρητής αυτός ονομάζεται flag_B και η τιμή του ανανεώνεται σε κάθε ακμή του ρολογιού που η είσοδος του συστήματος είναι έγκυρη. Ο πίνακας αληθείας του flag_B για κάθε έγκυρη χρονική στιγμή φαίνεται στον πίνακα 10. Με t συμβολίζεται ο τρέχον κύκλος ρολογιού που η είσοδος του κυκλώματος είναι έγκυρη. Αντίστοιχα με $t+1$ συμβολίζεται ο ακριβώς επόμενος κύκλος ρολογιού.

Πίνακας 10. Πίνακας αληθείας του σήματος flag_B.

flag_B _t	flag_B _{t+1}
0	1
1	0

Η τιμή του καταχωρητή flag_B και το σήμα εγκυρότητας της εισόδου στέλνονται σε μία ακόμα λογική πύλη XOR, όπου το αποτέλεσμα της απαρτίζει το σήμα εγκυρότητας της εξόδου B. Από την άλλη πλευρά, η έξοδος A είναι έγκυρη κάθε φορά που είναι έγκυρη και η αντίστοιχη είσοδος. Με αυτό το σύστημα το πρότυπο για την

αφαίρεση κάποιων bit κωδικοποιημένων δεδομένων είναι αυτό του σχήματος 11 που βρίσκεται στην ενότητα 2.3.3.

Στην τελευταία περίπτωση, όπου ο ρυθμός μετάδοσης είναι ίσος με 3/4, χρησιμοποιείται ένας μετρητής με μέγεθος δύο bit και ένας πολυπλέκτης. Σε κάθε ακμή του ρολογιού που η είσοδος είναι έγκυρη, ο μετρητής αυξάνει την τιμή του. Όταν φτάσει να έχει την τιμή “2” σε δεκαδική βάση και “10” σε δυαδική, επαναφέρεται στο μηδέν και συνεχίζει να αυξάνεται όπως και πριν. Ο πίνακας αληθείας του μετρητή φαίνεται στον πίνακα 11. Όπως και πριν με t συμβολίζεται ο τρέχον κύκλος ρολογιού που η είσοδος του κυκλώματος είναι έγκυρη, ενώ με $t+1$ συμβολίζεται ο ακριβώς επόμενος κύκλος ρολογιού.

Πίνακας 11. Πίνακα αληθείας του μετρητή και των σημάτων ελέγχου Enable_out_A και Enable_out_B.

Counter _t	Counter _{t+1}	Enable_out_A _{t+1}	Enable_out_B _{t+1}
00	01	1	1
01	10	1	0
10	00	0	1

Στην περίπτωση αυτή, η εγκυρότητα και των δύο σημάτων εξόδου εξαρτάται από τον μετρητή και τον πολυπλέκτη. Έτσι όποτε η είσοδος είναι έγκυρη, ο μετρητής θα ελέγχει την εγκυρότητα της εξόδου. Επομένως, αν η είσοδος δεν είναι έγκυρη, τότε δεν θα είναι έγκυρη ούτε και η έξοδος. Ο μετρητής ορίζει την τιμή του σήματος εγκυρότητας εξόδου ως εξής. Αν είναι ίσος με μηδέν, τότε και τα δύο σήματα εξόδου θα είναι έγκυρα. Αν είναι ίσος με ένα, τότε η έξοδος A θα είναι έγκυρη και η έξοδος B δεν θα είναι. Ενώ αν η τιμή του είναι ίση με δύο, τότε η έξοδος A δεν θα είναι έγκυρη ενώ η έξοδος B θα είναι. Η επιλογή αυτή γίνεται με την βοήθεια του πολυπλέκτη. Οι αντιστοιχίσεις των τιμών του μετρητή και των σημάτων εγκυρότητας εξόδου φαίνεται στον πίνακα 11, στον οποίο το πρώτο σήμα εγκυρότητας ονομάζεται “Enable_out_A” και το δεύτερο “Enable_out_B”. Όπως σε κάθε περίπτωση, έτσι και σε αυτή, το σύστημα παράγει το ίδιο πρότυπο με αυτό του πρωτοκόλλου, το οποίο φαίνεται στο σχήμα 12 της ενότητας 2.3.3.

4.9 Interleaver

Στην υλοποίησή του interleaver είναι σίγουρο ότι θα χρειαστεί να χρησιμοποιηθεί μία μνήμη ram ώστε να αποθηκευτεί και να διαβαστεί το block δεδομένων. Αυτή η ram θα διαθέτει δύο εισόδους για εγγραφή και δύο εξόδους για ανάγνωση. Αυτό σημαίνει ότι θα είναι διπλού ρυθμού δεδομένων (double data rate (DDR)). Χρησιμοποιείται μία τέτοια ram ώστε να αποφευχθεί η εγκατάσταση διάφορων buffer. Χρησιμοποιείται DDR RAM γιατί η είσοδος περιέχει δύο διαφορετικές τιμές που προέρχονται παράλληλα από το σύστημα του puncture. Αυτή η παραλληλία θα συνεχιστεί και στα δύο επόμενα κυκλώματα τον διαμορφωτή και το κύκλωμα συνένωσης του συμβόλου.

Για τις λειτουργίες της ανάγνωσης και της εγγραφής θα χρειαστούν επιπλέον οι διευθύνσεις μνήμης, η είσοδος του συστήματος αλλά και τα σήματα ενεργοποίησης της. Η μνήμη διαθέτει σύγχρονο σήμα επανεκκίνησης (reset), οπότε όταν αυτό ενεργοποιείται, τότε στην τρέχον ακμή του ρολογιού η μνήμη θα μηδενίζεται. Η εγγραφή στην μνήμη γίνεται μόνο όταν το σήμα ενεργοποίησης εγγραφής, για τουλάχιστον μία από τις εισόδους, είναι ενεργοποιημένο. Δηλαδή, κάθε χρονική στιγμή μπορεί να μην γίνεται εγγραφή, να γίνεται εγγραφή για την μία είσοδο ή να γίνεται εγγραφή και για τις δύο εισόδους ταυτόχρονα. Από την άλλη πλευρά, η ανάγνωση από την μνήμη γίνεται κάθε χρονική στιγμή που η έξοδος είναι έγκυρη, από τις θέσεις που υποδεικνύουν οι δύο διευθύνσεις ανάγνωσης. Η εγκυρότητά των σημάτων εξόδου διαχειρίζεται από ένα άλλο σύστημα που περιλαμβάνεται στην σχεδίαση του interleaver και περιγράφεται στην συνέχεια.

Πρέπει να τονιστεί ότι οι δύο εισοδοί δεδομένων της ram έχουν μέγεθος 1 bit η κάθε μία, ενώ οι έξοδοι δεδομένων έχουν μέγεθος 6 bit η κάθε μία. Η έξοδος της ram έχει αυτό το μέγεθος προκειμένου να καλύπτει κάθε τύπο διαμόρφωσης που θα ακολουθήσει. Έτσι ο interleaver διαβάζει το μέγιστο μέγεθος των έξι bit που απαιτείται από την 64-QAM διαμόρφωση και σε κάθε άλλη περίπτωση το κύκλωμα διαμόρφωσης θα απορρίψει

τα bit που δεν χρειάζεται. Στην συνέχεια της ενότητας θα περιγραφούν αναλυτικά τα υποσυστήματα διαχείρισης των διευθύνσεων ανάγνωσης και εγγραφής της μνήμης ram, το υποσύστημα διαχείρισης των δύο εισόδων και των αντίστοιχων σημάτων ενεργοποίησης εγγραφής, καθώς και το υποσύστημα διαχείρισης των σημάτων εξόδου και των σημάτων εγκυρότητας τους. Ο interleaver που υλοποιήθηκε απαρτίζεται από όλα τα παραπάνω υποσυστήματα.

Για τον υπολογισμό των νέων θέσεων των δεδομένων, το πρωτόκολλο ορίζει δύο σχέσεις. Όμως το τελικό αποτέλεσμα του συνδυασμού αυτών των δύο σχέσεων είναι γνωστό εξ' αρχής. Δηλαδή, είναι γνωστή η τελική θέση που πρέπει να αποθηκευτεί το κάθε ένα από τα bit δεδομένων, για κάθε είδος διαμόρφωσης που ζητείται από το σύστημα. Έτσι, με σκοπό την ελαχιστοποίηση της πολυπλοκότητας του κυκλώματος, δεν υπολογίζονται οι τελικές διευθύνσεις αλλά αυτές υπάρχουν αποθηκευμένες σε μνήμες rom. Τα block έχουν μέγεθος έως και 288 bit, οπότε θα χρειαστούν μνήμες rom έως και 288 bit για την αποθήκευση όλων των διευθύνσεων. Όμως παρατηρήθηκε ότι οι διευθύνσεις επαναλαμβάνονται ανά συγκεκριμένες θέσεις, μετατοπισμένες κατά συγκεκριμένες τιμές. Για παράδειγμα, στην περίπτωση όπου η διαμόρφωση είναι BPSK, οι διευθύνσεις εγγραφής επαναλαμβάνονται κάθε δεκαέξι θέσεις, αλλά οι διευθύνσεις κάθε νέας δεκαεξάδας εμφανίζονται αυξημένες κατά ένα. Οι τελικές διευθύνσεις που υπολογίζονται σύμφωνα με τις σχέσεις 6, 7 και 8, φαίνονται στον πίνακα 12 για κάθε διαμόρφωση που υποστηρίζει το σύστημα.

Πίνακας 12. Προϋπολογισμένες διευθύνσεις που παράγονται από τις σχέσεις 6, 7 και 8 της ενότητας 2.3.4.

Διαμόρφωση	Κωδικοποιημένα bits ανά σύμβολο (N_{CBPS})	Διευθύνσεις	Μετατόπιση
BPSK	48	0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 41, 44, 47	1
QPSK	96	0, 6, 12, 18, 24, 30, 36, 42, 48, 54, 60, 66, 72, 78, 84, 90, 1, 7, 13, 19, 25, 31, 37, 43, 49, 55, 61, 67, 73, 79, 85, 91, 2, 8, 14, 20, 26, 32, 38, 44, 50, 56, 62, 68, 74, 80, 86, 92, 3, 9, 15, 21, 27, 33, 39, 45, 51, 57, 63, 69, 75, 81, 87, 93, 4, 10, 16, 22, 28, 34, 40, 46, 52, 58, 64, 70, 76, 82, 88, 94, 5, 11, 17, 23, 29, 35, 41, 47, 53, 59, 65, 71, 77, 83, 89, 95	1
16-QAM	192	0, 13, 24, 37, 48, 61, 72, 85, 96, 109, 120, 133, 144, 157, 168, 181, 1, 12, 25, 36, 49, 60, 73, 84, 97, 108, 121, 132, 145, 156, 169, 180, 2, 15, 26, 39, 50, 63, 74, 87, 98, 111, 122, 135, 146, 159, 170, 183, 3, 14, 27, 38, 51, 62, 75, 86, 99, 110, 123, 134, 147, 158, 171, 182, 4, 17, 28, 41, 52, 65, 76, 89, 100, 113, 124, 137, 148, 161, 172, 185, 5, 16, 29, 40, 53, 64, 77, 88, 101, 112, 125, 136, 149, 160, 173, 184, 6, 19, 30, 43, 54, 67, 78, 91, 102, 115, 126, 139, 150, 163, 174, 187, 7, 18, 31, 42, 55, 66, 79, 90, 103, 114, 127, 138, 151, 162, 175, 186, 8, 21, 32, 45, 56, 69, 80, 93, 104, 117, 128, 141, 152, 165, 176, 189, 9, 20, 33, 44, 57, 68, 81, 92, 105, 116, 129, 140, 153, 164, 177, 188, 10, 23, 34, 47, 58, 71, 82, 95, 106, 119, 130, 143, 154, 167, 178, 191, 11, 22, 35, 46, 59, 70, 83, 94, 107, 118, 131, 142, 155, 166, 179, 190	2
64-QAM	288	0, 20, 37, 54, 74, 91, 108, 128, 145, 162, 182, 199, 216, 236, 253, 270, 1, 18, 38, 55, 72, 92, 109, 126, 146, 163, 180, 200, 217, 234, 254, 271, 2, 19, 36, 56, 73, 90, 110, 127, 144, 164, 181, 198, 218, 235, 252, 272, 3, 23, 40, 57, 77, 94, 111, 131, 148, 165, 185, 202, 219, 239, 256, 273, 4, 21, 41, 58, 75, 95, 112, 129, 149, 166, 183, 203, 220, 237, 257, 274, 5, 22, 39, 59, 76, 93, 113, 130, 147, 167, 184, 201, 221, 238, 255, 275,	3

		6, 26, 43, 60, 80, 97, 114, 134, 151, 168, 188, 205, 222, 242, 259, 276, 7, 24, 44, 61, 78, 98, 115, 132, 152, 169, 186, 206, 223, 240, 260, 277, 8, 25, 42, 62, 79, 96, 116, 133, 150, 170, 187, 204, 224, 241, 258, 278, 9, 29, 46, 63, 83, 100, 117, 137, 154, 171, 191, 208, 225, 245, 262, 279, 10, 27, 47, 64, 81, 101, 118, 135, 155, 172, 189, 209, 226, 243, 263, 280, 11, 28, 45, 65, 82, 99, 119, 136, 153, 173, 190, 207, 227, 244, 261, 281, 12, 32, 49, 66, 86, 103, 120, 140, 157, 174, 194, 211, 228, 248, 265, 282, 13, 30, 50, 67, 84, 104, 121, 138, 158, 175, 192, 212, 229, 246, 266, 283, 14, 31, 48, 68, 85, 102, 122, 139, 156, 176, 193, 210, 230, 247, 264, 284, 15, 35, 52, 69, 89, 106, 123, 143, 160, 177, 197, 214, 231, 251, 268, 285, 16, 33, 53, 70, 87, 107, 124, 141, 161, 178, 195, 215, 232, 249, 269, 286, 17, 34, 51, 71, 88, 105, 125, 142, 159, 179, 196, 213, 233, 250, 267, 287	
--	--	--	--

Γνωρίζοντας αυτή την περιοδικότητα των τιμών, χρησιμοποιούνται τέσσερις μνήμες rom όπου αποθηκεύονται μόνο τα βασικά πρότυπα των διευθύνσεων, τα οποία φαίνονται με πιο έντονη γραμματοσειρά στον ίδιο πίνακα .

Για την εξαγωγή των διευθύνσεων από τις rom χρησιμοποιείται ένας μετρητής, ο οποίος αυξάνεται ανά δύο ή ανά μία τιμή, ανάλογα με το πόσες από τις δύο εισόδους είναι έγκυρη. Δηλαδή αν δεν είναι καμία είσοδος έγκυρη τότε δεν θα αυξηθεί καθόλου, αν είναι μόνο η μία είσοδος έγκυρη θα αυξηθεί κατά ένα, ενώ αν είναι και οι δύο έγκυρες τότε θα αυξηθεί κατά δύο. Έτσι θα ληφθεί η κατάλληλη τιμή από το πρότυπο διευθύνσεων της κάθε rom. Επιπλέον, χρησιμοποιείται ένα ακόμα σύστημα με μετρητές και πολυπλέκτες, που υπολογίζει την μετατόπιση των διευθύνσεων από τα βασικά πρότυπα. Το μέγεθος της μετατόπισης φαίνεται και αυτό στον πίνακα 12, για κάθε δυνατή διαμόρφωση. Κάθε χρονική στιγμή που μία τουλάχιστον από τις εισόδους είναι έγκυρη υπολογίζεται μία νέα διεύθυνση εγγραφής της μνήμης ram, η οποία προκύπτει από την πρόσθεση μεταξύ της διεύθυνσης που δίνουν οι μνήμες rom και της μετατόπισης.

Για την υποστήριξη των διαφορετικών συνδυασμών ρυθμού κώδικα και διαμόρφωσης, όλα τα παραπάνω κυκλώματα λειτουργούν με διάφορους παραμέτρους που ελέγχουν την σωστή ροή των δεδομένων. Έτσι υπάρχει ένα ακόμα σύστημα που χειρίζεται τις παραμέτρους αυτές. Συνοψίζοντας λέμε ότι το τμήμα του interleaver που διαχειρίζεται τις διευθύνσεις μνήμης για την εγγραφή των δεδομένων χρησιμοποιεί τέσσερις μνήμες rom, έναν μετρητή, ένα σύστημα που διαχειρίζεται την μετατόπιση και ένα σύστημα που διαχειρίζεται τις παραμέτρους.

Όπως είπαμε και προηγουμένως, η διεύθυνσεις μνήμης που δίνουν οι rom, δείχνουν στις δύο επόμενες θέσεις όταν λάβει δύο έγκυρες εισόδους. Ενώ αν μόνο μία από τις εισόδους είναι έγκυρη, τότε η rom θα δείχνει μόνο μία νέα διεύθυνση. Δηλαδή, αν στο σύστημα εισαχθεί μόνο ένα έγκυρο σήμα, τότε θα χρειαστεί μία από τις διευθύνσεις που διαθέτει η rom. Οπότε η δεύτερη διεύθυνση δεν θα χρησιμοποιηθεί. Έτσι αν στο επόμενο κύκλο ρολογιού εισαχθούν δύο έγκυρα σήματα, τότε η προηγούμενη περισσευούμενη διεύθυνση θα χρησιμοποιηθεί τώρα. Επιπλέον, θα χρειαστεί να χρησιμοποιηθεί και η δεύτερη καινούργια διεύθυνση που διαθέτει η rom. Συμπεραίνουμε, ότι θα χρησιμοποιείται μόνο η πρώτη ή και οι δύο διευθύνσεις ανά περίπτωση. Άρα όταν θα είναι διαθέσιμες δύο εισόδοι, τότε αυτές θα στέλνονται αυτούσιες στην μνήμη ram. Όμως άμα είναι μόνο μία από τις δύο εισόδους διαθέσιμη, τότε όποια και αν είναι αυτή, πρέπει να σταλθεί στην πρώτη είσοδο δεδομένων της ram, μίας και η έγκυρη διεύθυνση που δίνει η rom στέλνεται στην πρώτη είσοδο διεύθυνσης της ram. Αυτή ακριβώς είναι η λειτουργία του υποσυστήματος διαχείρισης εισόδου και των αντίστοιχων σημάτων εγκυρότητας.

Ο τρόπος για να επιτευχθεί η λειτουργία αυτή είναι απλός. Αρχικά υπολογίζεται η πρώτη είσοδος της ram και μετά το πρώτο σήμα εγκυρότητας. Έτσι στην πρώτη είσοδο δεδομένων της ram στέλνεται το σήμα εισόδου “Α”, αν το αντίστοιχο σήμα εγκυρότητας είναι ενεργό, αλλιώς στέλνεται το σήμα “Β”. Σε όλες τις περιπτώσεις η δεύτερη είσοδος δεδομένων της ram θα τροφοδοτείται με το σήμα “Β”. Ενώ, για τα σήματα εγκυρότητας εισόδου της ram, το πρώτο bit ισούται με το αποτέλεσμα της λογικής πράξης OR ανάμεσα στα δύο bit του σήματος εγκυρότητας της εισόδου του κυκλώματος και το δεύτερο bit ισούται με το αποτέλεσμα της λογικής πράξης AND ανάμεσα στα ίδια σήματα. Για τον σωστό συγχρονισμό και τη σωστή ροή δεδομένων στον interleaver τα σήματα εισόδου και τα αντίστοιχα σήματα εγκυρότητας ολισθαίνουν κατά δύο κύκλους ρολογιού.

Με αυτή την μέθοδο η ram συνεργάζεται σωστά με τα συστήματα διαχείρισης των διευθύνσεών της. Ο πίνακας αληθείας για τις εισόδους και τα σήματα εγκυρότητας της ram φαίνονται στον πίνακα 13.

Πίνακας 13. Πίνακας αληθείας για τις εισόδους και τα σήματα εγκυρότητας της ram.

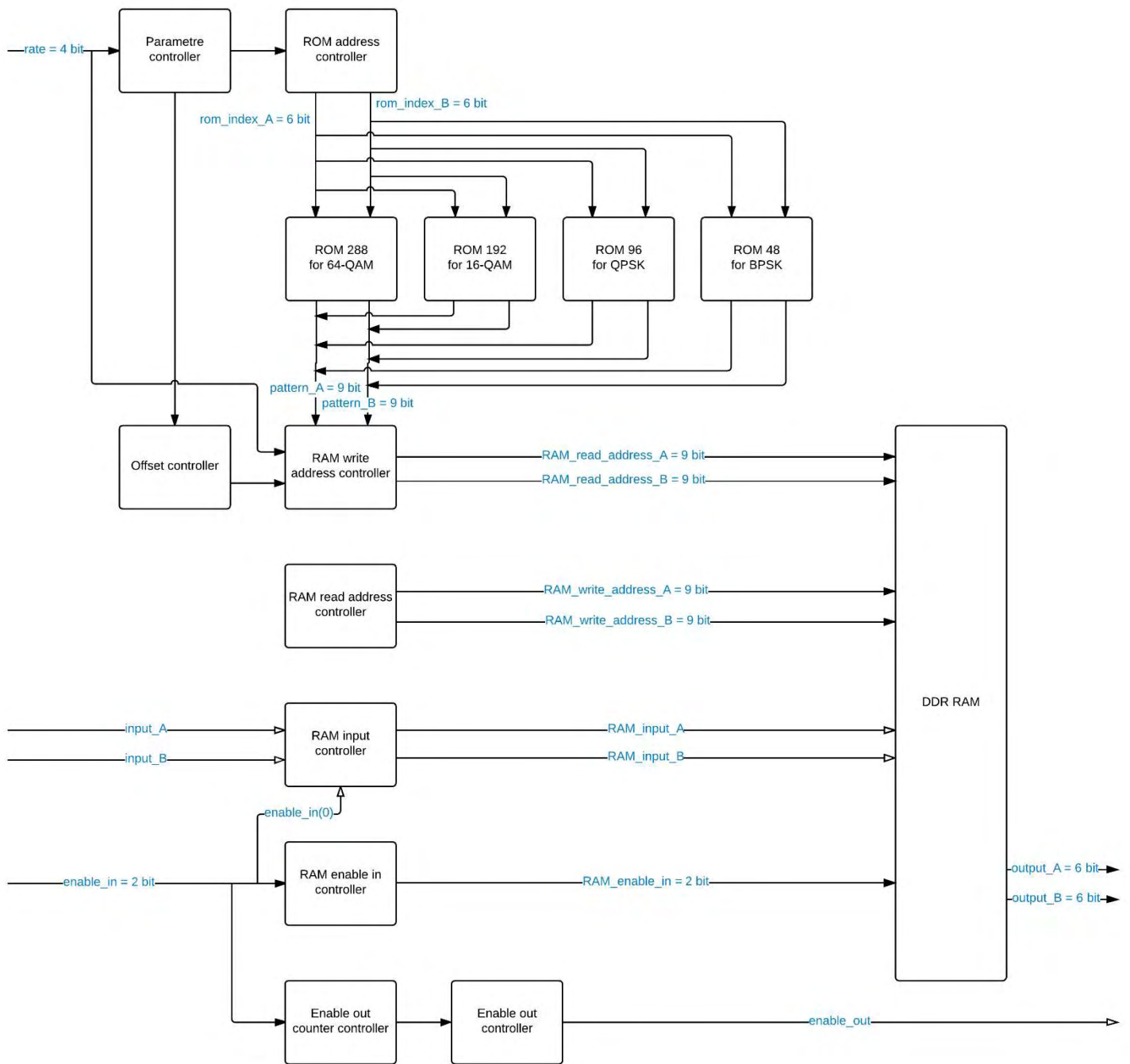
enable_in_B	enable_in_A	ram_input_B	ram_input_A	ram_enable_B	ram_enable_A
0	0	input_B	input_A	0	0
0	1	input_b	input_A	0	1
1	0	input_B	input_B	0	1
1	1	input_B	input_A	1	1

Εξ' ίσου απλή είναι και η λειτουργία του υποσυστήματος διαχείρισης των σημάτων εξόδου και των αντίστοιχων σημάτων εγκυρότητας. Το σήμα εγκυρότητας της εξόδου ενεργοποιείται μόλις αποθηκευτούν δεκαέξι σύνολα μεγέθους N_{BPSC} στην ram και απενεργοποιείται μετά από οχτώ κύκλου ρολογιού. Η καταμέτρηση των κύκλων ρολογιού, όπου το σήμα εγκυρότητας εξόδου θα είναι ενεργοποιημένο, γίνεται από έναν μετρητή. Όσο το σήμα εγκυρότητας της εξόδου είναι ενεργό πρέπει να παράγεται και η σωστή έξοδος. Για τον σκοπό αυτό πρέπει να παράγονται οι σωστές διευθύνσεις ανάγνωσης για την μνήμη ram.

Οι διευθύνσεις ανάγνωσης παράγονται με την βοήθεια του σήματος εγκυρότητας της εξόδου, ενός ακόμα μετρητή και την τιμή μετατόπισης που λαμβάνει από το αντίστοιχο υποσύστημα. Σε κάθε ακμή του ρολογιού όπου το σήμα εγκυρότητας εξόδου είναι ενεργό, ο μετρητής αυξάνεται. Στην τιμή του μετρητή προστίθεται η κατάλληλη μετατόπιση και το αποτέλεσμα που προκύπτει αποτελεί την διεύθυνση ανάγνωσης της μνήμης ram. Αφού το σήμα εγκυρότητας εξόδου θα είναι ενεργό για 16 κύκλους ρολογιού, τότε και ο μετρητής όταν φτάσει να είναι ίσος με 16 πρέπει να μηδενιστεί και να συνεχίσει από το μηδέν όταν ξαναχρημαστεί. Οι τιμές της διεύθυνσης ανάγνωσης παράγονται με τρόπο που επιτρέπουν πρώτα να ολοκληρωθεί η εγγραφή μίας συγκεκριμένης τιμής στην μνήμη και στην συνέχεια να εκτελεσθεί η ανάγνωσή της. Η ανάγνωση από την μνήμη γίνεται σε κάθε κύκλο ρολογιού, όμως η έξοδος είναι έγκυρη μόνο όταν υποδεικνύεται από το αντίστοιχο σήμα εγκυρότητας.

Με διπλό ρυθμό ανάγνωση από την ram, μπορούν να διαβαστούν 16 τιμές σε 8 κύκλους ρολογιού. Έτσι, για την ανάγνωση ενός ολόκληρου συμβόλου, μεγέθους 48 τέτοιων τιμών, θα χρειαστούν τρεις επαναλήψεις ανάγνωσης. Η κάθε μία από αυτές τις 48 τιμές θα σταλθεί στο κύκλωμα διαμόρφωσης, ώστε να αντιστοιχηθεί σε μία συγκεκριμένη συχνότητα. Όπως αναφέραμε και παραπάνω, το μέγεθος των σημάτων εξόδου είναι 6 bit, προκειμένου να καλύπτεται κάθε είδος διαμόρφωσης που μπορεί να ζητηθεί.

Πρέπει να σημειωθεί ότι αν ενωθούν με την σειρά όλα τα δεδομένα που διαβάστηκαν από την μνήμη ram, δεν θα είναι στην σειρά που υποδεικνύει το πρωτόκολλο. Αυτό συμβαίνει γιατί γράφονται με την σωστή σειρά στην μνήμη, δηλαδή όπως ακριβώς ορίζει το πρωτόκολλο, αλλά δεν τα διαβάζονται με τον ίδιο τρόπο. Προκειμένου να διαβάζονται τα δεδομένα γρήγορα και τελικά, να προσδίδεται καθυστέρηση μόνο ενός κύκλου ρολογιού στο συνολικό σύστημα, ο interleaver χρησιμοποιείται μόνο για να λαμβάνονται σωστά οι τιμές που αντιστοιχούν στο κάθε υποφέρον σήμα του συμβόλου, μεγέθους N_{BPSC} (με μέγιστο μέγεθος τα 6 bit). Έτσι τα δεδομένα βγαίνουν με την ίδια λανθασμένη σειρά και από το κύκλωμα διαμόρφωσης. Τελικά, διορθώνεται το λάθος αυτό και τα δεδομένα μπαίνουν στην σωστή σειρά, χωρίς να αυξηθεί η καθυστέρηση, στο κύκλωμα όπου ενώνονται όλες οι τιμές συχνοτήτων των υποφέρον σημάτων σε ένα σύμβολο. Το κύκλωμα αυτό λέγεται symbol former και περιγράφεται στην ενότητα 2.3.5. Τέλος, για την ευκολότερη κατανόηση της δομής του interleaver που υλοποιήθηκε δίνεται το σχηματικό του διάγραμμα, το οποίο φαίνεται στο σχήμα 49.



Σχήμα 49. Σχηματικό διάγραμμα του Interleaver.

4.10 Κύκλωμα διαμόρφωσης

Για την υλοποίησή του συστήματος χαρτογράφησης των τιμών συχνοτήτων χρησιμοποιείται η τεχνική look up table. Αφού έχουν προϋπολογιστεί οι τιμές των μιγαδικών αριθμών που προκύπτουν από την σχέση 9 της ενότητας 2.3.5, θα διαβάζεται η είσοδος και θα αντιστοιχίζεται σε μία συγκεκριμένη τιμή από αυτές. Όπως είπαμε και στην ενότητα 4.9, η έξοδος του interleaver θα είναι 6 bit προκειμένου να μπορεί να εξυπηρετήσει και τα τέσσερα μεγέθη του N_{BPSK} . Για την ακρίβεια η έξοδος του interleaver διαθέτει δύο αριθμούς των 6 bit. Άρα και η είσοδος του συστήματος διαμόρφωσης θα λαμβάνει αυτές τις δύο εισόδους μεγέθους 6 bit, ώστε να

συνεχιστεί η παραλληλία και να μην επιτραπεί η χρήση buffer. Αφού θα λαμβάνει δύο τιμές που χρειάζονται διαμόρφωση και προκειμένου να μην αυξηθεί η καθυστέρηση του συστήματος, υλοποιούνται παράλληλα δυο συστήματα διαμόρφωσης, ένα για κάθε μία από τις εισόδους. Επομένως, θα εξάγει αυτές τις δύο διαμορφωμένες τιμές προς το σύστημα του symbol former. Για την σωστή ροή δεδομένων θα χρειαστεί να ενσωματωθεί και το σήμα εγκυρότητας της εξόδου, το οποίο θα απαρτίζεται από το σήμα εγκυρότητας εισόδου ολισθημένο κατά ένα κύκλο ρολογιού. Χρησιμοποιείται το ίδιο σήμα εγκυρότητας εξόδου και για τις δύο εξόδους, μιας και τα δύο σήματα θα είναι έγκυρα ταυτόχρονα.

Τα σήματα εξόδου είναι μιγαδικοί αριθμοί, των οποίων τα πραγματικά και τα φανταστικά μέρη είναι κλασματικοί αριθμοί. Για την εύκολη εκτέλεση των πράξεων μεταξύ κλασματικών αριθμών χρησιμοποιείται συγκεκριμένη αναπαράσταση. Η αναπαράσταση των κλασματικών αριθμών που χρησιμοποιήθηκε στην παρούσα εργασία εξηγείται στο παράρτημα 8.2. Ενώ ο τρόπος διαχείρισης του φανταστικού και του πραγματικού τμήματος ενός μιγαδικού αριθμού περιγράφεται στο παράρτημα 8.1. Η κάθε μία από τις εξόδους αποτελείται από δύο 18 bit αριθμούς, έναν για το πραγματικό μέρος και έναν για το φανταστικό μέρος του μιγαδικού αριθμού. Αυτό συμβαίνει πρώτον, γιατί είναι δύο μιγαδικοί, όπου διαθέτουν και πραγματικό και φανταστικό μέρος. Έτσι όπως ορίζεται και στο παράρτημα 8.1, θα χρειαστούν 4 διαφορετικά σήματα για την αποθήκευση όλων. Δεύτερον, γιατί η μεγαλύτερη δυνατή τιμή που μπορεί να προκύψει από το σύστημα διαμόρφωση είναι η 1.080123. Έτσι σύμφωνα με το παράρτημα 8.1, θα χρησιμοποιηθεί το αποτέλεσμα της σχέσης 15.

Σχέση 14

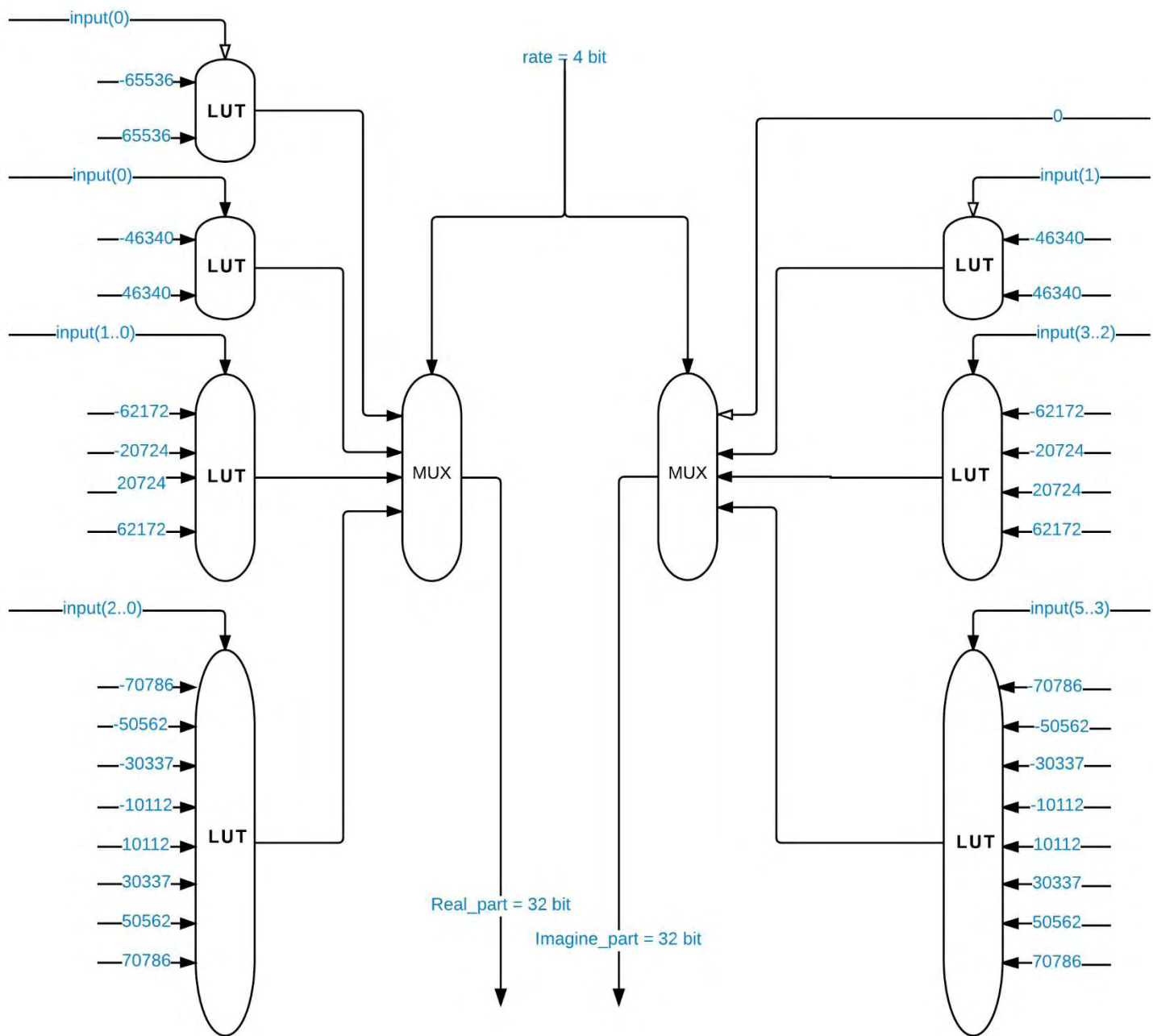
$$1.080123 * 65536 = 70786.970401$$

Σχέση 15

$$\text{Floor}(70786.970401) = 70786$$

Όπου ο αριθμός αυτός χρειάζεται 17 bit για να αναπαρασταθεί συν 1 bit για το πρόσημο. Συνολικά θα χρειαστούν 18 bit για να υποστηριχθεί και η πιο μεγάλη τιμή που μπορεί να χρησιμοποιηθεί.

Τελικά, η συνολική έξοδος του συστήματος αποτελείται από τέσσερις 18 bit αριθμούς και από το αντίστοιχο σήμα εγκυρότητα της εξόδου. Είναι σημαντικό να τονιστεί ότι η είσοδος μπορεί να έχει μέγεθος 6 bit, όμως τα σημαντικά δεδομένα μπορεί να περιορίζονται μόνο στο πρώτο bit από αυτά (ή σε περισσότερα από 1, όπως 4 και 2). Το πόσα bit της εισόδου είναι σημαντικά, εξαρτάται πλήρως από τον επιλεγόμενο τύπο διαμόρφωσης. Όπως αναφέραμε και στην ενότητα 2.3.5 ο τύπος διαμόρφωσης επιλέγεται βάσει του ζητούμενου ρυθμού δεδομένων. Το σχηματικό διάγραμμα του διαμορφωτή φαίνεται στο σχήμα 50. Υπενθυμίζουμε ότι θα υλοποιηθούν δύο τέτοια κυκλώματα, ένα για κάθε μία από τις εισόδους.



Σχήμα 50. Σχηματικό διάγραμμα του κυκλώματος διαμόρφωσης.

4.11 Κύκλωμα συνένωσης του συμβόλου

Στην υλοποίηση του symbol former, για την αποθήκευση των δεδομένων χρησιμοποιούνται μνήμες ram. Για την ακρίβεια οι μνήμες είναι δύο. Στην πρώτη αποθηκεύονται τα πραγματικά μέρη των δεδομένων, ενώ στην δεύτερη αποθηκεύονται τα φανταστικά μέρη. Όπως είχε αναφερθεί στην ενότητα 4.10, το σύστημα διαμόρφωσης στέλνει δύο σήματα στον symbol former. Έτσι οι ram που θα χρησιμοποιηθούν θα είναι διπλού ρυθμού δεδομένων, ώστε να μπορούν να αποθηκεύουν ταυτόχρονα και τις δύο τιμές που λαμβάνουν και για να διατηρηθεί η παραλληλία. Η εγγραφή στις μνήμες γίνεται κάθε φορά που η είσοδος είναι έγκυρη. Όπως τα δεδομένα έτσι και οι πιλοτικές τιμές αποθηκεύονται σε κατάλληλες θέσεις των μνημών. Αυτές όμως αποθηκεύονται κατά την εκκίνηση του συστήματος ή μετά την ολοκλήρωση της αποθήκευσης του κάθε

συνόλου δεδομένων. Οι πιλοτικές τιμές λαμβάνονται απευθείας από το υποσύστημα διαχείρισης τους, το οποίο ενσωματώνεται μέσα στο κύκλωμα του symbol former.

Οι διευθύνσεις μνήμης που πρόκειται να αποθηκευτούν τα δεδομένα παράγονται από έναν μετρητή. Ο μετρητής αυξάνεται κάθε φορά που η είσοδος είναι έγκυρη και όταν φτάσει να είναι ίσος με 23, αν η είσοδος είναι έγκυρη για μία ακόμη φορά, τότε μηδενίζεται. Με τα ίδια κριτήρια που μηδενίζεται ο μετρητής ενεργοποιείται και το σήμα ολίσθησης της εγκυρότητας της εξόδου. Δηλαδή αν έχουν διαβαστεί και οι 48 αριθμοί και το κύκλωμα λάβει μία ακόμα έγκυρη είσοδο, τότε ενεργοποιείται το σήμα ολίσθησης της εγκυρότητας της εξόδου. Το σήμα ολίσθησης της εγκυρότητας της εξόδου χρησιμοποιείται και για την ενεργοποίηση της εγγραφής στους καταχωρητές εξόδου του κυκλώματος. Έτσι στον επόμενο κύκλο ρολογιού από αυτόν που ενεργοποιήθηκε, οι τιμές της ram αποθηκεύονται στους καταχωρητές πριν σταλούν στην έξοδο του συστήματος. Μετά από ακόμα έναν κύκλο ρολογιού τα δεδομένα θα φτάσουν στην έξοδο και ταυτόχρονα το σήμα εγκυρότητας της εξόδου θα ενεργοποιηθεί, μιας και έλαβε την τιμή του καθυστερημένα λόγω της ολίσθησης. Η αντιγραφή των τιμών της ram στους καταχωρητές εξόδου γίνεται προκειμένου να αλλάξει η σειρά των δεδομένων πριν αποσταλούν στο κύκλωμα του αντίστροφου γρήγορου μετασχηματισμού Fourier.

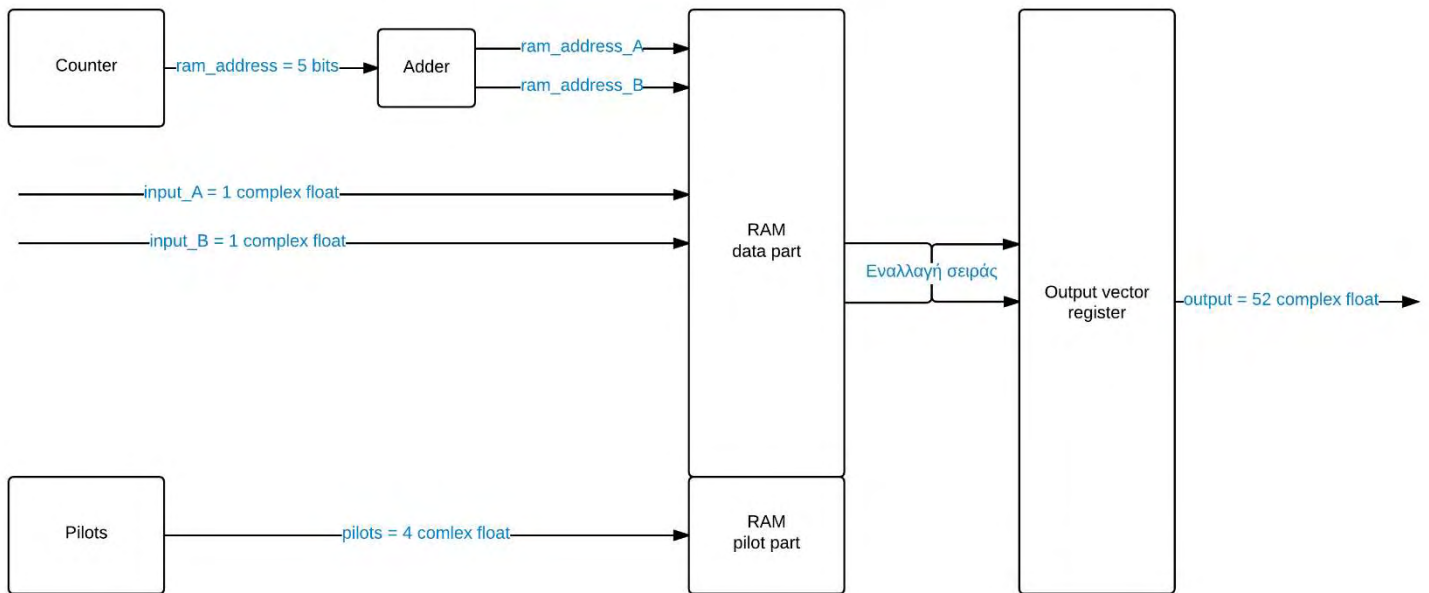
Όπως είχαμε πει στην ενότητα 4,9, ο interleaver δεν στέλνει τα δεδομένα στο σύστημα διαμόρφωσης με την σωστή σειρά. Έτσι και ο symbol former τα λαμβάνει με την ίδια λανθασμένη σειρά. Λαμβάνοντας υπόψη αυτή την σειρά και γνωρίζοντας την σειρά που το κύκλωμα του αντίστροφου γρήγορου μετασχηματισμού Fourier πρέπει να λάβει τα δεδομένα, βάσει του πρωτοκόλλου, υπολογίσαμε τον τρόπο με τον οποίο πρέπει να αντιγραφτούν τα δεδομένα από την μνήμη στους καταχωρητές εξόδου. Ο τρόπος εξαγωγής των σημάτων φαίνεται στον πίνακα 14, όπου στην πρώτη στήλη φαίνεται η σειρά εισόδου των δεδομένων στο κύκλωμα, στην δεύτερη στήλη φαίνεται η μετονομασμένη σειρά που ορίζει το πρωτόκολλο, στην τρίτη στήλη φαίνεται η ζητούμενη σειρά των δεδομένων από το κύκλωμα του αντίστροφου γρήγορου μετασχηματισμού Fourier, στην τέταρτη στήλη φαίνεται η σειρά που τα δεδομένα λαμβάνονται, από τον interleaver, και αποθηκεύονται στην μνήμη και στην πέμπτη στήλη φαίνεται η τελική τροποποιημένη σειρά των δεδομένων, η οποία αντιστοιχεί στην ζητούμενη σειρά του αντίστροφου γρήγορου μετασχηματισμού Fourier.

Πίνακας 14. Αρίθμηση των τιμών του συμβόλου..

Αρχική αρίθμηση των δεδομένων	Τελική αρίθμηση των δεδομένων	Ζητούμενη σειρά από τον IFFT	Σειρά δεδομένων μέσα στην μνήμη	Αντιστοίχιση δεδομένων μνήμης στην έξοδο
1	-26	1	0	39
2	-25	2	3	23
3	-24	3	6	7
4	-23	4	9	38
5	-22	5	12	22
6	-21	6	15	6
7	-20	7	18	Pilot (-7)
8	-19	8	21	37
9	-18	9	24	21
10	-17	10	27	5
11	-16	11	30	36
12	-15	12	33	20
13	-14	13	36	4
14	-13	14	39	35
15	-12	15	42	19
16	-11	16	45	3
17	-10	17	1	34
18	-9	18	4	18

19	-8	19	7	2
20	-7	20	10	33
21	-6	21	13	Pilot (-21)
22	-5	22	16	17
23	-4	23	19	1
24	-3	24	22	32
25	-2	25	25	16
26	-1	26	28	0
27	1	-26	31	47
28	2	-25	34	31
29	3	-24	37	15
30	4	-23	40	46
31	5	-22	43	30
32	6	-21	46	Pilot (21)
33	7	-20	2	14
34	8	-19	5	45
35	9	-18	8	29
36	10	-17	11	13
37	11	-16	14	44
38	12	-15	17	28
39	13	-14	20	12
40	14	-13	23	43
41	15	-12	26	27
42	16	-11	29	11
43	17	-10	32	42
44	18	-9	35	26
45	19	-8	38	10
46	20	-7	41	Pilot (7)
47	21	-6	44	41
48	22	-5	47	25
49	23	-4	Pilot (-21)	9
50	24	-3	Pilot (-7)	40
51	25	-2	Pilot (7)	24
52	26	-1	Pilot (21)	8

Με τον τρόπο που υποδεικνύεται από την πέμπτη στήλη, υλοποιούνται όλες οι αλλαγές που ορίζει το πρωτόκολλο σε έναν κύκλο ρολογιού. Το σχηματικό διάγραμμα του symbol former φαίνεται στο σχήμα 51.

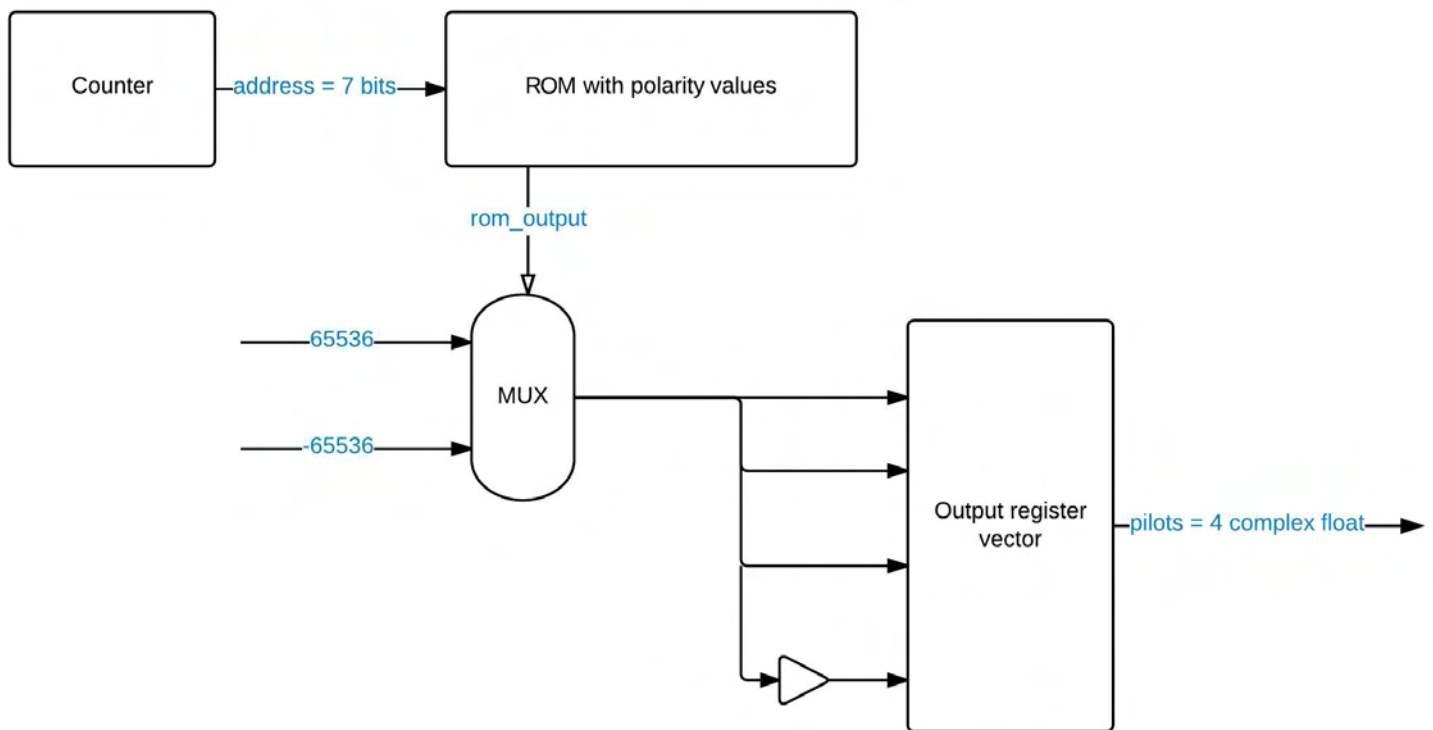


Σχήμα 51. Σχηματικό διάγραμμα του κυκλώματος συνένωσης του συμβόλου.

4.12 Κύκλωμα παραγωγής πιλοτικών τιμών

Στην υλοποίηση του κυκλώματος παραγωγής πιλοτικών τιμών έχει αποθηκευτεί το σύνολο πολικότητας σε μία μνήμη rom μεγέθους 127 bit. Το σύνολο πολικότητας φαίνεται στην σχέση 11 της ενότητας 2.3.7. Τα θετικά στοιχεία του συνόλου έχουν αποθηκευτεί ως άσσοι, ενώ τα αρνητικά στοιχεία έχουν αποθηκευτεί ως μηδενικά. Η ανάκτηση των τιμών της μνήμης rom γίνεται με την βοήθεια ενός μετρητή, ο οποίος αυξάνεται κατά την εκκίνηση της εισαγωγή των δεδομένων στο συνολικό σύστημα και κάθε φορά που πρέπει να ενοποιηθεί ένα νέο σύμβολο. Ο μετρητής αυτός επαναφέρεται στο μηδέν όταν η τιμή του φτάσει να είναι ίση με το 127. Ενώ η εξαγωγή των κατάλληλων πιλοτικών τιμών γίνεται με την βοήθεια ενός πολυπλέκτη. Έτσι αν η τιμή που υποδεικνύει το σύνολο πολικότητας είναι άσσος, τότε οι πιλοτικές τιμές θα είναι 1, 1, 1 και -1 και θα σταλούν στην έξοδο με την ίδια σειρά. Ενώ αν η τιμή που υποδεικνύει το σύνολο πολικότητας είναι μηδενική, τότε οι πιλοτικές τιμές θα είναι -1, -1, -1 και 1, μιας και θα είναι οι αρχικές πιλοτικές τιμές πολλαπλασιασμένες με το -1.

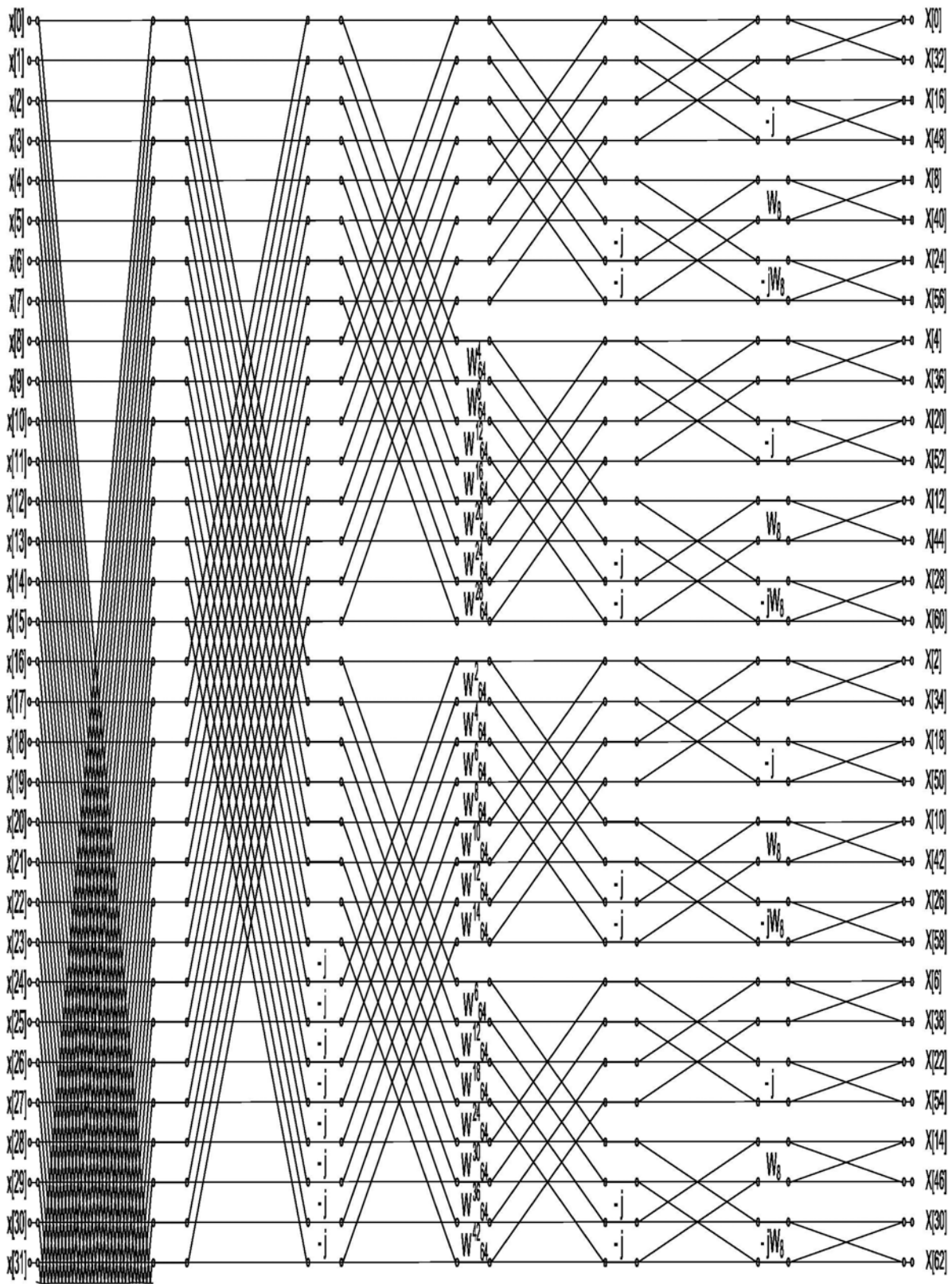
Σημειώνουμε ότι η κάθε πιλοτική τιμή έχει μόνο πραγματικό μέρος, έτσι κατά την παραλαβή τους από το σύστημα του symbol former, το φανταστικό του μέρος θα ορίζεται ως μηδενικό. Επιπλέον, το μέγεθός τους θα είναι 18 bit και η αναπαράστασή τους θα είναι ίδια με αυτή των διαμορφωμένων σημάτων πληροφορίας. Οι πιλοτικές τιμές που εξάγονται είναι έγκυρες μόνο για έναν κύκλο ρολογιού κατά την εκκίνηση της εισαγωγής των δεδομένων στο συνολικό σύστημα και μετά την συνένωση ενός νέου συμβόλου. Η εγκυρότητα αυτή ορίζεται από το αντίστοιχο σήμα εγκυρότητας της εξόδου. Το σχηματικό διάγραμμα του κυκλώματος φαίνεται στο σχήμα 52.



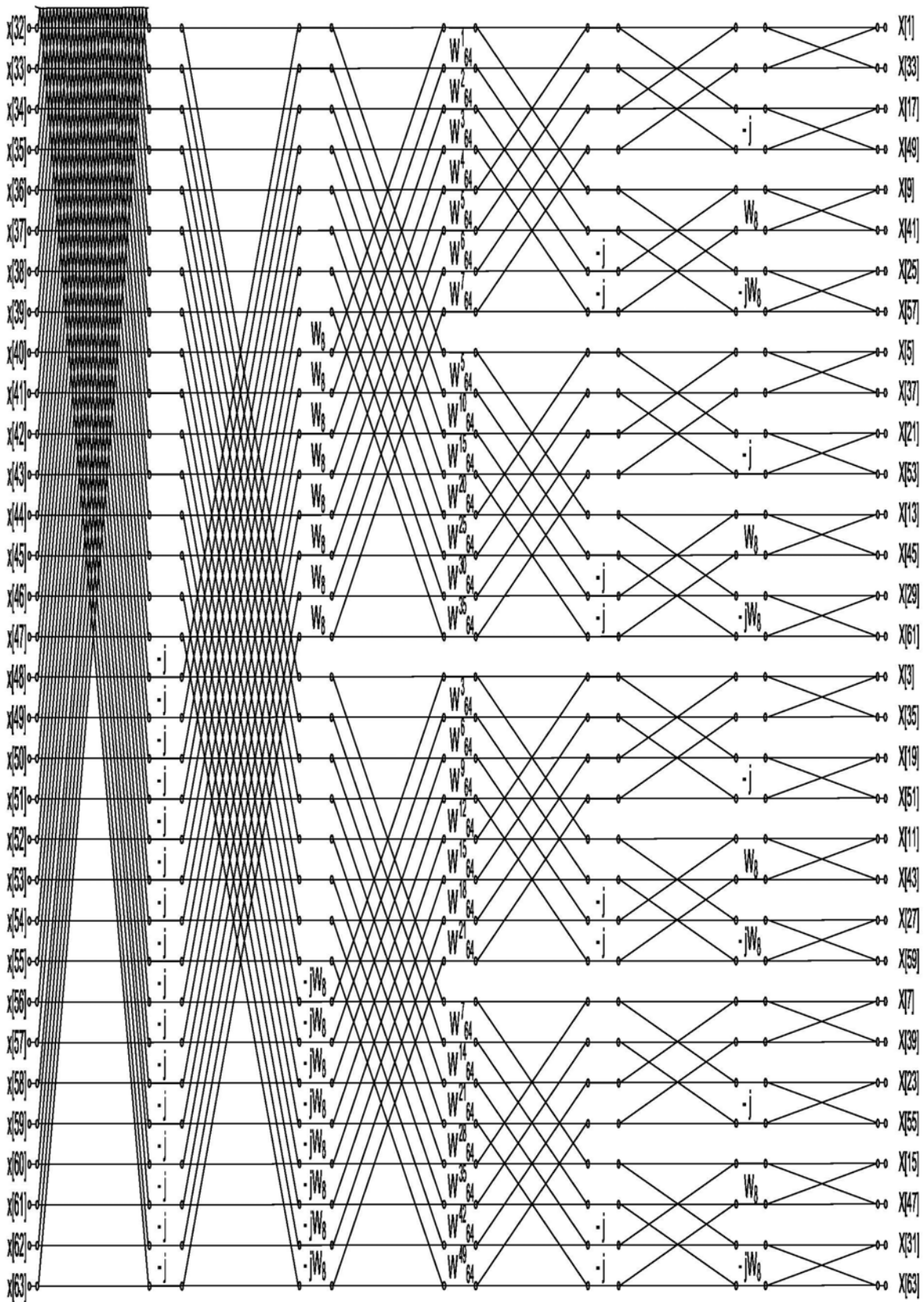
Σχήμα 52. Σχηματικό διάγραμμα κυκλώματος παραγωγής πιλοτικών τιμών.

4.13 Αντίστροφος γρήγορος μετασχηματισμός Fourier

Στο κύκλωμα του αντίστροφου γρήγορου μετασχηματισμού Fourier που υλοποιήθηκε, έχει χρησιμοποιηθεί μία σχεδίαση από μία πατέντα, η οποία χρησιμοποιεί λιγότερες πράξεις από την κλασική σχεδίαση του. Η σχεδίαση του φαίνεται αναλυτικά στο σχήματα 53 και 54, τα οποία ανήκουν και αυτά στην αναφορά της πατέντας [8].



Σχήμα 53. Σχηματικό διάγραμμα του αντίστροφου γρήγορου μετασχηματισμού Fourier (μέρος α).



Σχήμα 54. Σχηματικό διάγραμμα του αντίστροφου γρήγορου μετασχηματισμού Fourier (μέρος β).

Το σχήμα αυτό απεικονίζει έναν αντίστροφο γρήγορο μετασχηματισμό Fourier όπου τα αποτελέσματα του κάθε σταδίου εξάγονται παράλληλα. Για την ικανοποίηση του κρίσιμου μονοπατιού χρησιμοποιείται επικάλυψη (pipeline) στο κύκλωμα. Το κρίσιμο μονοπάτι έχει οριστεί εξαρχής ως το μονοπάτι ενός πολλαπλασιαστή και ενός προσθέτη στην σειρά. Για να επιτευχθεί η επικάλυψη πριν από το κάθε στάδιο υπάρχει μία ομάδα από θέσεις μνήμης. Το κύκλωμα αυτό αποτελείται από 6 στάδια, όπου το κάθε ένα αντιστοιχεί σε ένα από τα επίπεδα δικτύου πεταλούδας (butterfly network) 64, 32, 16, 8, 4 και 2 της σχεδίασης. Έτσι 6 θα είναι και τα στάδια της επικάλυψης.

Όμως, επειδή υπάρχουν πράξεις με μιγαδικούς αριθμούς, σε κάποια από αυτά τα 6 στάδια θα απαιτούνται περισσότερες πράξεις στην σειρά από αυτές που καλύπτει το προορισμένο κρίσιμο μονοπάτι, ώστε να παραχθεί το ζητούμενο αποτέλεσμα. Η μεγαλύτερη σειρά από πράξεις που μπορεί να προκύψουν είναι ένα πολλαπλασιασμός και δύο προσθέσεις. Για να αποτραπεί η επέκταση του κρίσιμου μονοπατιού, “σπάνε” σε δύο τμήματα τα στάδια που απαιτούν τέτοιου είδους πράξεις. Για την κατανόηση της απαιτούμενης αλλαγής δίνεται ένα παράδειγμα στο παράρτημα 8.3, όπου φαίνονται οι αναγκαίες πράξεις αλλά και η υπέρβαση του μήκους του ζητούμενου κρίσιμου μονοπατιού. Τα στάδια του κυκλώματος που χρειάζονται να υπολογίσουν πιο σύνθετες πράξεις είναι τα 32, 16 και 4. Τελικά, αφού θα χωριστούν στην μέση τα 3 από τα 6 στάδια, τότε συνολικά τα στάδια επικάλυψης που θα απαρτίζουν τον αντίστροφο γρήγορο μετασχηματισμό Fourier θα είναι 9.

Για να μην υλοποιηθούν τόσο πολλοί καταχωρητές και αριθμητικές πράξεις, όσοι χρειάζεται η παράλληλη σχεδίαση του IFFT που φαίνεται να χρειάζονται στα σχήματα 53 και 54, η δομή του κυκλώματος θα διαφέρει λίγο. Συγκεκριμένα, οι θέσεις μνήμης που θα χωρίζουν τα στάδια της επικάλυψης θα είναι buffers και οι αριθμητικές πράξεις που απαιτούνται δεν θα εκτελούνται ταυτόχρονα, αλλά θα εκτελούνται σειριακά. Η επικάλυψη της αρχικής σχεδίασης θα εξακολουθεί να υπάρχει, έτσι το κύκλωμα του IFFT θα λειτουργεί με τον τρόπο που ορίζεται από την πατέντα [8]. Αυτό που θα αλλάξει είναι ότι το πρώτο αποτέλεσμα του IFFT θα εξαγεται μερικούς κύκλους ρολογιού αργότερα, από ότι θα εξαγόταν στην παράλληλη σχεδίαση της πατέντας. Ενώ η πολυπλοκότητα του κυκλώματος θα πέσει αρκετά, τόσο ώστε να μην ενοχλεί η καθυστέρηση που προσδίδει η αλλαγή αυτή.

Στο πρώτο στάδιο, το οποίο ονομάζεται στάδιο 64, θα υπολογίζονται κάθε χρονική στιγμή 4 αποτελέσματα, ταυτόχρονα. Στην συνέχεια, το στάδιο 32 θα λαμβάνει αυτά τα 4 αποτελέσματα και θα παράγει άλλα 4. Όπως αναφέρθηκε και παραπάνω, το στάδιο αυτό θα είναι “σπασμένο” στην μέση με ένα καταχωρητή, ώστε να διατηρηθεί το κρίσιμο μονοπάτι, το ίδιο ισχύει και για τα στάδια 16 και 4. Τα 4 αποτελέσματα θα περνούν σε 4 διαφορετικούς buffers 8 θέσεων, ένα στον κάθε έναν. Το στάδιο 32 θα χρειαστεί $64/4 = 16$ κύκλους ρολογιού ώστε να επεξεργαστεί όλα τα δεδομένα του συμβόλου. Έτσι για να εκτελεί άμεσα τις αριθμητικές πράξεις το στάδιο 16, στους buffers του σταδίου 32 θα αποθηκεύονται τα αποτελέσματα μόνο στους 8 πρώτους κύκλους ρολογιού, ενώ στους υπόλοιπους 8 θα αποστέλλονται απευθείας και με την σειρά. Έτσι μετά από 8 κύκλους ρολογιού το στάδιο 16 θα λαμβάνει ταυτόχρονα τους 8 αριθμούς που χρειάζεται. Στην συνέχεια το στάδιο 16 θα εξάγει ταυτόχρονα 8 αποτελέσματα, τα οποία θα αποθηκεύονται σε 8 buffers 4 θέσεων. Μετά από δύο κύκλους ρολογιού το στάδιο 8 θα λαμβάνει τους 16 αριθμούς, και θα παράγει 16. Αντίστοιχα θα αποθηκεύει τα αποτελέσματα σε 16 buffers 2 θέσεων, από τους οποίους το στάδιο 4 θα τα διαβάζει κάθε δύο κύκλους ρολογιού. Το στάδιο 4 θα παράγει 32 αριθμούς κάθε χρονική στιγμή, όπου μία φορά θα τα αποθηκεύει σε μία σειρά από καταχωρητές και μία θα τα στέλνει απευθείας στο στάδιο 2. Τέλος το στάδιο 2 θα λαμβάνει ταυτόχρονα τις 64 τιμές που αναμένονται και θα εξάγει και τις 64 τιμές του συμβόλου στον επόμενο κύκλο ρολογιού, ταυτόχρονα.

Οι συντελεστές του μετασχηματισμού Fourier που απαιτούνται από το κάθε στάδιο, είναι αποθηκευμένοι και αυτοί σε διανύσματα καταχωρητών. Σύμφωνα με την πατέντα το κάθε στάδιο χρειάζεται διαφορετικούς συντελεστές ή δεν χρειάζεται κάποιον από αυτούς [8]. Έτσι για την αποθήκευση των αναγκαίων συντελεστών χρησιμοποιείται μία μνήμη rom και μερικοί καταχωρητές. Το στάδιο 32 απαιτεί ένα μεγάλο πλήθος συντελεστών και είναι αυτό που χρησιμοποιεί την rom με τους συντελεστές. Ενώ τα άλλα δύο στάδια πεταλούδας, το 16 και το 4, δεν απαιτούν πολλούς συντελεστές, έτσι χρησιμοποιούνται μόνο κάποιοι καταχωρητές για την αποθήκευσή τους. Σημειώνουμε ότι το κάθε στοιχείο μνήμης που περιέχει συντελεστές

βρίσκεται στο εσωτερικό του υποκυκλώματος του κάθε σταδίου. Επομένως, η μνήμη rom βρίσκεται στο εσωτερικό της σχεδίασης του σταδίου 32, ενώ οι καταχωρητές συντελεστών βρίσκονται στα στάδια 16 και 4.

Πρέπει να αναφέρουμε ότι τόσο οι συντελεστές όσο και τα δεδομένα είναι φανταστικοί αριθμοί, έτσι αποτελούνται από πραγματικό και από φανταστικό μέρος. Για την αναπαράστασή τους χρησιμοποιείται η ίδια τεχνική με τα προηγούμενα κυκλώματα, η αναπαράσταση αυτή περιγράφεται στα παραρτήματα 8.1. Όποτε τα μέρη του κάθε φανταστικού αριθμού θα έχουν μέγεθος 18 bit στο στάδιο 64. Όμως θα εκτελεστούν πράξεις πρόσθεσης, όπου στην χειρότερη περίπτωση θα χρειαστεί να προστεθούν οι δύο μεγαλύτεροι αριθμοί που μπορούν να προκύψουν. Έτσι οι τιμές στην έξοδο του σταδίου 64 θα χρειαστεί να είναι μεγαλύτερες κατά ένα bit ώστε να μην χαθεί πληροφορία λόγω υπερχειλίσσης (overflow). Δηλαδή θα πρέπει να έχουν μέγεθος 19 bit. Σημειώνεται ότι θα υπάρχουν και πράξεις πολλαπλασιασμού, αλλά ο πολλαπλασιαστής θα είναι ένας από τους συντελεστές του μετασχηματισμού. Αυτό σημαίνει ότι θα είναι μικρότερος της μονάδας, γιατί θα είναι η τιμή κάποιου ημιτόνου ή συνημίτονου. Έτσι δεν υπάρχει περίπτωση να αυξήσει το μέγεθος της τιμής πληροφορίας. Στην συνέχεια θα χρειάζεται να προστίθεται ένα ακόμα bit σε κάθε επόμενο στάδιο για τον ίδιο λόγο. Έτσι οι τιμές των σταδίων 32, 16, 8 και 4 θα είναι 20, 21, 22 και 23. Η έξοδος του σταδίου 2 θα διαφέρει, γιατί οι τιμές του πριν εξαχθούν διαιρούνται με τον αριθμό 64. Οπότε, αντί να απαιτούνται 24 bit θα χρειάζονται μόνο 18 bit, όσα ακριβώς εισάγονται και στο στάδιο 64.

Η είσοδος του κυκλώματος θα έπρεπε να ζητάει 64 φανταστικούς αριθμούς, όμως εμείς γνωρίζουμε ότι κάποια συγκεκριμένα σήματα, από την είσοδο που ζητάει, θα είναι μηδενικά. Έτσι στην είσοδο του κυκλώματος στέλνουμε μόνο την χρήσιμη πληροφορία, η οποία προέρχεται από το κύκλωμα symbol former και αποτελείται από συχνότητες που αντιστοιχούν σε 48 τιμές δεδομένων και 4 πιλοτικές τιμές. Η σωστή αντιστοίχιση των σημάτων εισόδου έχει αναφερθεί στην ενότητα 2.3.8 και φαίνεται στο σχήμα 17. Οπότε η συνολική είσοδος του κυκλώματος θα έχει μέγεθος 52 σημάτων και 935 bit, όπως φαίνεται και στην σχέση 16.

Σχέση 16

$$52 \text{ υποφέρον σήματα} * (18 * 2) \text{ μέγεθος μιγαδικού αριθμού σε bit} = 935 \text{ bit}$$

Οι πράξεις που απαιτούσαν την είσοδο των μηδενικών στοιχείων υλοποιήθηκαν διαφορετικά. Πλέον θέτουν την τιμή του κατάλληλου σήματος ίση με το μηδέν και υπολογίζουν το αποτέλεσμα απευθείας.

Τέλος σημειώνεται ότι η έξοδος του κυκλώματος θα έχει μέγεθος 1151 bit και θα απαρτίζεται από 64 σήματα φανταστικών αριθμών, όπως φαίνεται και στην σχέση 17.

Σχέση 17

$$64 \text{ υποφέρον σήματα} * (18 * 2) \text{ μέγεθος μιγαδικού αριθμού σε bit} = 1151 \text{ bit}$$

Η έξοδος αυτή στέλνεται απευθείας στο κύκλωμα εισαγωγής κυκλικού προθέματος και παραθύρωσης, του οποίου η υλοποίηση περιγράφεται στην επόμενη ενότητα (ενότητα 4.14).

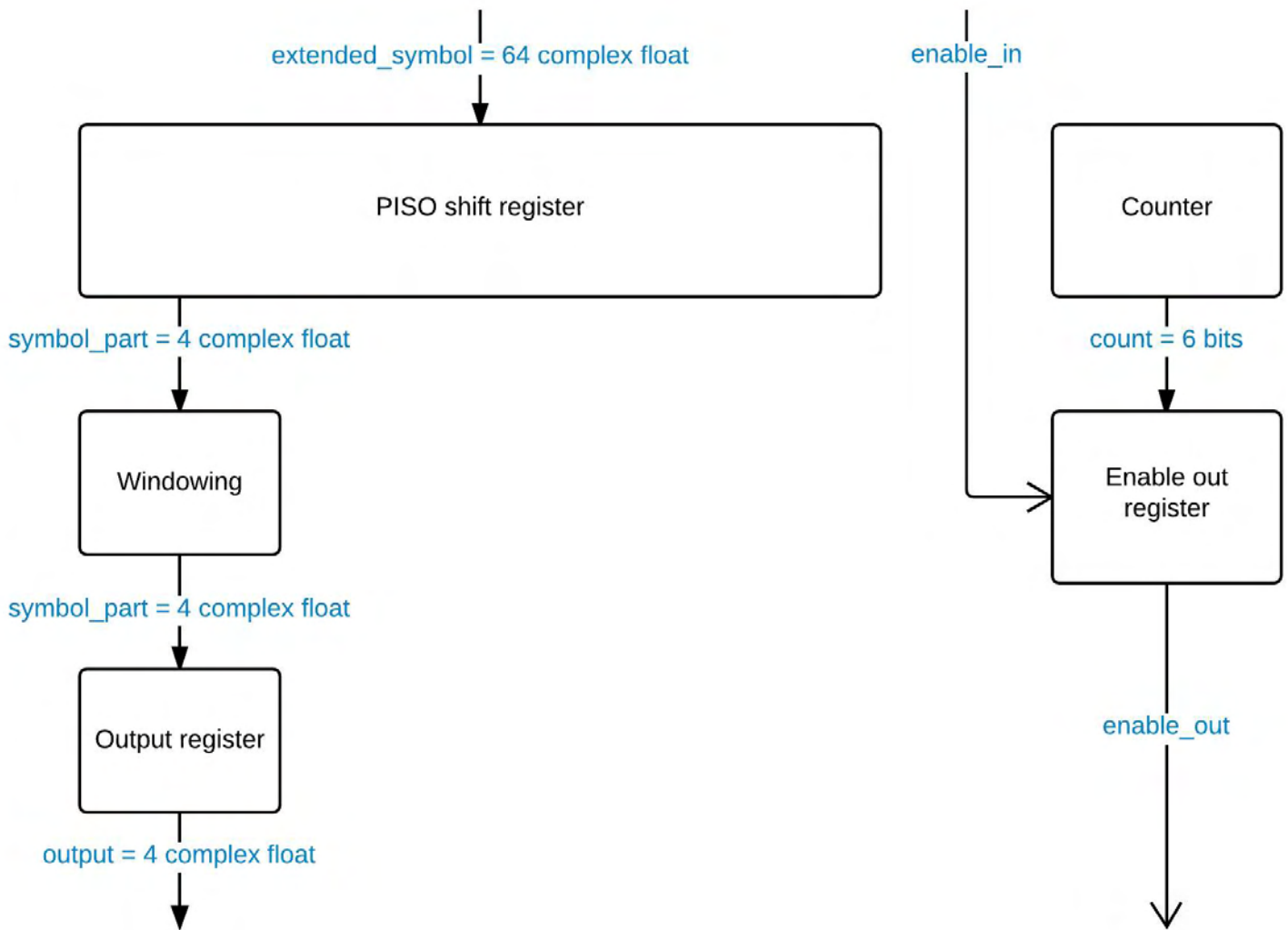
4.14 Εισαγωγή τιμών φύλαξης και παραθύρωσης

Στο κύκλωμα εισαγωγής τιμών φύλαξης και παραθύρωσης (Guard Interval (GI) and Windowing) έχει προσαρμοστεί η διαφορά του μεγέθους του κυκλικού προθέματος που απαιτείται ανά περίπτωση. Για την επιλογή του επιθυμητού κυκλικού προθέματος χρησιμοποιούνται κάποια σήματα ελέγχου, τα οποία το κύκλωμα τα λαμβάνει ως είσοδο από την μηχανή καταστάσεων. Σε κάθε περίπτωση, κατά την εισαγωγή έγκυρης εισόδου, σε έναν βοηθητικό καταχωρητή αποθηκεύεται το υποδιπλάσιο της πρώτης τιμής του συμβόλου προκειμένου να είναι διαθέσιμο στο επόμενο σύμβολο που θα τη χρειαστεί. Αυτός ο καταχωρητής αρχικά θα είναι μηδενισμένος. Ενώ αυτή η υποδιπλασιασμένη τιμή αποθηκεύεται για χάρη της λειτουργίας του windowing.

Στον ίδιο κύκλο ρολογιού και με την ίδια συνθήκη εγγραφής, αποθηκεύονται στον καταχωρητή εξόδου του συστήματος οι 4 πρώτες τιμές του προθέματος που προσκομίζονται από το σύμβολο, ώστε να αποσταλούν

πρώτες στο επίπεδο ελέγχου πρόσβασης πολυμέσων. Τονίζουμε ότι οι τιμές αυτές υπάρχουν και μέσα στο ίδιο το σύμβολο. Επιπλέον, πριν αποσταθούν στην έξοδο του συστήματος, η πρώτη από τις τέσσερις τιμές του προθέματος υποδιπλασιάζεται και στην συνέχεια αυξάνεται κατά την τιμή του βοηθητικού καταχωρητή. Κατά τον υπολογισμό του πρώτου συμβόλου ο βοηθητικός καταχωρητής θα είναι μηδενισμένος, άρα η πρώτη τιμή που θα εξαχθεί θα είναι αυτούσια η υποδιπλασιασμένη πρώτη τιμή του προθέματος. Ταυτόχρονα στον ίδιο κύκλο ρολογιού, το υπόλοιπο του προθέματος και οι τιμές του συμβόλου, από την αρχή του έως το σημείο όπου αρχίζουν οι ίδιες τιμές με αυτές του υπόλοιπου προθέματος, αποθηκεύονται σε έναν καταχωρητή ολίσθησης παράλληλης εισόδου με την σωστή σειρά. Το σύμβολο στην είσοδο του προέρχεται παράλληλα από την έξοδο του αντίστροφου γρήγορου μετασχηματισμού Fourier.

Για κάθε επόμενο κύκλο ρολογιού μέχρι να ολοκληρωθεί η εξαγωγή του σήματος, αυτός ο καταχωρητής ολισθαίνει ανά τέσσερις θέσεις. Επιπλέον, οι τέσσερις πρώτες τιμές του καταχωρητή ολίσθησης αποθηκεύονται στον καταχωρητή εξόδου. Με αυτόν τον τρόπο θα εξαχθεί το προσδοκώμενο πρόθεμα ακολουθούμενο από το κατάλληλο σύμβολο. Η διαδικασία ολίσθησης των τιμών διαρκεί διαφορετικό αριθμό κύκλων ρολογιού, για κάθε περίπτωση. Έτσι μπορεί να παραχθεί είτε το πρότυπο που αναμένεται για τα πεδία signal και data, είτε το πρότυπο για τα τμήματα του preamble. Όμως σε κάθε περίπτωση στην έξοδο στέλνονται οι τέσσερις τιμές του καταχωρητή εξόδου. Για τη εκγυροποίηση της εξόδου του συστήματος, χρησιμοποιείται ένας μετρητής, ο οποίος αυξάνεται σε κάθε κύκλο ρολογιού μετά την είσοδο ενός συμβόλου. Προκειμένου να υποστηριχθούν όλες οι περιπτώσεις, ο μετρητής έχει διαφορετικές τερματικές τιμές. Με την βοήθεια αυτού του μετρητή παράγεται το σωστό σήμα εγκυρότητας της εξόδου για κάθε δυνατή περίπτωση. Το σχηματικό διάγραμμα του κυκλώματος του GI και windowing φαίνεται στο σχήμα 55.



Σχήμα 55. Σχηματικό διάγραμμα του κυκλώματος εισαγωγής τιμών φύλαξης και παραθύρωσης.

5 Προσομοίωση και επαλήθευση της σχεδίασης

Στο κεφάλαιο αυτό θα περιγράψει η διαδικασία προσομοίωσης και επαλήθευσης που ακολουθήθηκε, ώστε να εξακριβωθεί η σωστή λειτουργία του κυκλώματος που σχεδιάστηκε και καλείται να υλοποιηθεί. Η διαδικασία αυτή αποτελείται από δύο τμήματα. Από την μία πλευρά είναι η προσομοίωση, για την οποία χρησιμοποιείται ένα testbench και το εργαλείο προσομοίωσης ModelSim της Altera. Ενώ από την άλλη πλευρά χρησιμοποιείται ένα πρόγραμμα σε C, το οποίο εκτελεί σημασιολογικά την ζητούμενη λειτουργία. Αυτό σημαίνει ότι το πρόγραμμα σε C δεν έχει την δομή που έχει και η σχεδίαση για το υλικό, όμως εκτελεί την ίδια λειτουργία.

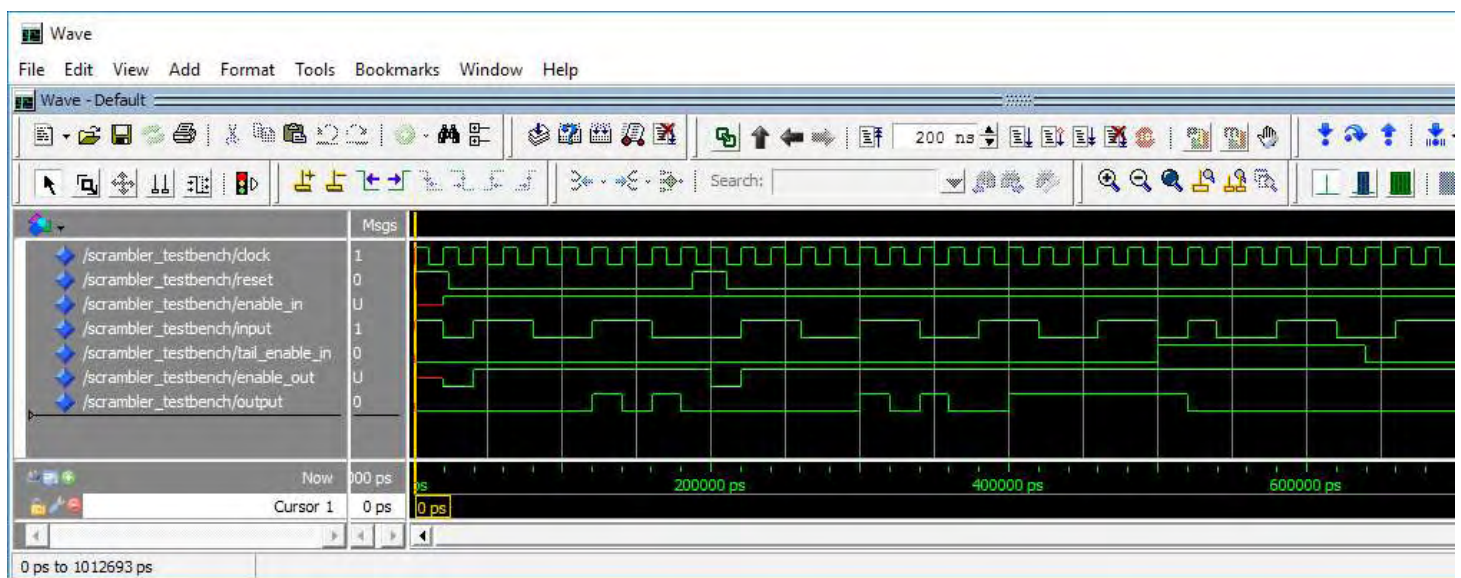
Για το κάθε ένα από τα υλοποιημένα υποκυκλώματα, αλλά και συνολικά για ολόκληρη την σχεδίαση του πομπού, χρησιμοποιείται ένα testbench, όπως είπαμε και πριν. Με την βοήθεια αυτών των testbench προσομοιώνεται το κύκλωμα και αποθηκεύονται οι τιμές της εξόδου του συστήματος σε ένα αρχείο, για κάθε κύκλο ρολογιού. Στην συνέχεια το αρχείο αυτό δίνεται ως είσοδος στο λογισμικό τμήμα. Το πρόγραμμα σε C ελέγχει την ομοιότητα μεταξύ των τιμών του αρχείου και των τιμών που το ίδιο παρήγαγε. Αν οι τιμές αυτές δεν ήταν ίδιες τότε θα υπήρχε κάποιο πρόβλημα στην λειτουργία του κυκλώματος. Πράγμα που θα οδηγούσε στην επανεξέταση και στην τροποποίηση των σχεδιάσεων. Όμως αν οι τιμές του λογισμικού και του υλικού είναι ίδιες, τότε αυτό σημαίνει ότι το κύκλωμα λειτουργεί επιθυμητά. Μετά την επαλήθευση της σωστής λειτουργίας

του κάθε υποκυκλώματος, ελέγχθηκε και ολόκληρο το κύκλωμα με ακριβώς τον ίδιο τρόπο. Στο υπόλοιπο του κεφαλαίου θα περιγραφεί ένα παράδειγμα που περιέχει τις πληροφορίες γύρω από την προσομοίωση στο ModelSim και γύρω από την επαλήθευση με την χρήση των προγραμμάτων σε C.

Στο παράδειγμα αυτό θα περιγράψουμε την διαδικασία προσομοίωσης και επαλήθευσης του scrambler, όπου το κύκλωμά του ήταν το πρώτο που σχεδιάστηκε. Μετά την ολοκλήρωση της σχεδίασής του και της συγγραφής του κώδικά του σε γλώσσα VHDL, συγγράφηκε το testbench που βοήθησε στην προσομοίωση και στην αποθήκευση των αποτελεσμάτων της σε αρχείο. Το testbench γράφεται και αυτό σε VHDL, στο οποίο ορίζεται η περιοδικότητα του σήματος του ρολογιού. Σύμφωνα με το σήμα του ρολογιού υπολογίζεται το σήμα επανεκκίνησης (reset), το σήμα εισόδου, το σήμα εγκυρότητας της εισόδου και το σήμα που υποδεικνύει ότι η είσοδος ανήκει στο πεδίο tail του data. Δηλαδή, σε κάθε ακμή του ρολογιού τα σήματα αυτά μπορούν και αλλάζουν την τιμή τους. Αυτό επιτυγχάνεται με την χρήση καταχωρητών για το κάθε ένα. Το σήμα reset ενεργοποιείται για έναν κύκλο ρολογιού κατά την εκκίνηση του κυκλώματος και μετά από δέκα κύκλους ρολογιού. Η καταμέτρηση των κύκλων ρολογιού γίνεται με την βοήθεια ενός μετρητή. Το σήμα reset υπολογίζεται με αυτόν τον τρόπο ώστε να εξακριβωθεί η σωστή λειτουργία του. Πρέπει να πούμε πως το σήμα reset αλλάζει τιμές σύμφωνα με το ρολόι επειδή το αντίστοιχο σήμα μέσα στο κύκλωμα του scrambler είναι σύγχρονο. Αν δεν υπολογιζόταν με αυτόν τον τρόπο, τότε μπορεί να υπήρχαν αλλαγές στις τιμές του καταχωρητή του που δεν θα ήταν επιθυμητές.

Από την άλλη πλευρά για τον υπολογισμό της εισόδου, η οποία πρέπει να είναι ψευδοτυχαία ώστε να φανερωθούν τυχόν σφάλματα στην λειτουργία του scrambler που δεν λήφθηκαν υπόψιν κατά την σχεδίασή του, περνιούνται κάποιες από τις τιμές του ήδη υπάρχον μετρητή από μία λογική πύλη XOR. Πιο συγκεκριμένα, για τον υπολογισμό της τιμής της εισόδου περνιέται το πρώτο, το δεύτερο και το πέμπτο bit του μετρητή από μία λογική πύλη XOR τριών εισόδων. Το σήμα εγκυρότητας της εισόδου ενεργοποιείται σε κάθε κύκλο ρολογιού που υπάρχει διαθέσιμη μία τιμή εισόδου. Ενώ το σήμα tail_enable_in ενεργοποιείται για 6 κύκλους ρολογιού μετά από 16 κύκλους από την εκκίνηση του κυκλώματος.

Μετά την ολοκλήρωση του testbench, αυτό και την σχεδίαση του scrambler εισάγονται στο ModelSim. Για να το γίνει αυτό πρέπει πρώτα να δημιουργηθεί ένα νέο project, μετά να εισαχθούν τα δύο αρχεία, να μεταγλωττιστούν και τελικά να εκκινήσει η προσομοίωση. Ένα στιγμιότυπο από την προσομοίωση του scrambler φαίνεται στο σχήμα 56.



Σχήμα 56. Στιγμιότυπο από την προσομοίωση του Scrambler με το ModelSim.

Η σχεδίαση του scrambler δεν είναι μεγάλη, έτσι όταν ολοκληρωθεί η προσομοίωση ελέγχονται τα δεδομένα εξόδου με το μάτι, ώστε αν παρατηρηθεί σφάλμα, να μην εκκινήσει το επόμενο βήμα. Αν ξεκινούσε

απευθείας το επόμενο βήμα θα καθυστερούσε λίγο περισσότερο όλη διαδικασία επαλήθευσης. Μετά από μία σειρά αλλαγών τα αποτελέσματα είναι ικανοποιητικά και εκκινείται ο έλεγχος με την βοήθεια του λογισμικού. Για την λειτουργία του λογισμικού απαιτείται η είσοδος του αρχείου που παράχθηκε κατά την προσομοίωση. Το αρχείο αυτό αποτελείται από τρεις στήλες μία για το σήμα εγκυρότητας της εισόδου, μία για την ίδια την είσοδο και μία για την έξοδο. Αν το σήμα reset έχει ενεργοποιηθεί τότε αυτό φαίνεται με την προσθήκη μίας επιπλέον σειράς με τον αριθμό 2. Υπενθυμίζουμε ότι το αρχείο αυτό παράγεται κατά την προσομοίωση του scrambler από το testbench. Ένα δείγμα του αρχείου φαίνεται στο σχήμα 57.

1	2
2	1 0 0
3	1 1 0
4	1 1 0
5	1 0 0
6	1 0 0
7	1 1 1
8	1 1 0
9	1 0 1
10	1 0 0
11	2
12	1 0 0
13	1 1 0
14	1 1 0
15	1 0 0
16	1 0 0

Σχήμα 57. Στιγμιότυπο από το αρχείο που παράχθηκε κατά την προσομοίωση του Scrambler.

Στο λογισμικό υλοποιείται ο scrambler όπως ακριβώς ορίζεται από το πρωτόκολλο. Για την επαλήθευση της σωστής λειτουργίας του λογισμικού, συγκρίνονται οι τιμές που παράγονται με τις τιμές που δίνουν τα παραδείγματα που αναφέρονται στο πρωτόκολλο, για την ίδια είσοδο. Το κεφάλαιο που περιέχει το παράδειγμα που χρησιμοποιήθηκε λέγεται Annex L [1]. Όταν εκκινηθεί το λογισμικό, αυτό αρχίζει και διαβάζει δεδομένα από το αρχείο που του δόθηκε. Αν συναντήσει την τιμή 2, τότε επανεκκινείται και ο αντίστοιχος scrambler του λογισμικού. Εάν η πρώτη στήλη διαθέτει έναν άσος, τότε σημαίνει πως το σήμα εισόδου είναι έγκυρο. Μετά διαβάζει το έγκυρο σήμα εισόδου και το σήμα εξόδου και αν το αποτέλεσμα που το λογισμικό παραγάγει είναι ίδιο με την τιμή που διαβάστηκε από την τρίτη στήλη, τότε το κύκλωμα λειτουργεί σωστά. Η διαδικασία αυτή συνεχίζεται έως ότου ολοκληρωθεί η ανάγνωση ολόκληρου του αρχείου. Αν δεν βρεθεί διαφορά ανάμεσα στις τιμές εξόδους του υλικού και του λογισμικού scrambler, τότε αυτό σημαίνει ότι ο υλικός scrambler θα είναι ίδιος με αυτόν που ορίζει το πρωτόκολλο. Σε κάθε άλλη περίπτωση ελέγχουμε την σχεδίαση του υλικού scrambler για να ανιχνεύσουμε το λάθος.

Ένα στιγμιότυπο από την διαδικασία επαλήθευσης του scrambler με τη βοήθεια του προγράμματος σε C φαίνεται στο σχήμα 58.

```

Energopoih8hke to reset.

Shma metasxhmatismo: 1011101
input: 0, hardware_out: 0, software_out: 0

Shma metasxhmatismo: 0101110
input: 1, hardware_out: 0, software_out: 0

Shma metasxhmatismo: 1010111
input: 1, hardware_out: 0, software_out: 0

Shma metasxhmatismo: 1101011
input: 0, hardware_out: 0, software_out: 0

Shma metasxhmatismo: 0110101
input: 0, hardware_out: 1, software_out: 1

Shma metasxhmatismo: 1011010
input: 1, hardware_out: 0, software_out: 0

Shma metasxhmatismo: 1101101
input: 1, hardware_out: 1, software_out: 1

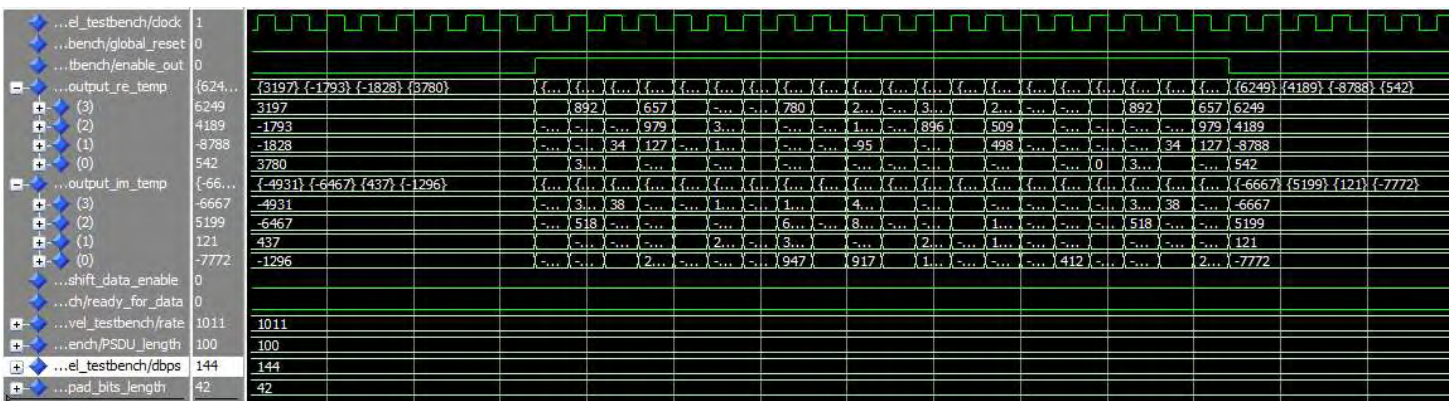
Energopoih8hke to reset.

Shma metasxhmatismo: 1011101
input: 0, hardware_out: 0, software_out: 0

```

Σχήμα 58. Στιγμιότυπο από την επιβεβαίωση της σωστής λειτουργίας του Scrambler με την βοήθεια προγράμματος σε C.

Στην εικόνα αυτή διακρίνεται η ομοιότητα μεταξύ των τιμών εξόδου του υλικού και του λογισμικού για ένα μικρό πλήθος επαναλήψεων. Όμως η ομοιότητα αυτή ισχύει για όσες επαναλήψεις και αν εκτελεστεί ο έλεγχος. Έτσι ολοκληρώνεται η διαδικασία προσομοίωσης και επαλήθευσης του scrambler και συνεχίζουμε στο επόμενο κύκλωμα. Εδώ παρουσιάσαμε την διαδικασία αυτή μόνο για τον scrambler, όμως στην πραγματικότητα την εκτελέσαμε για κάθε ένα από τα κυκλώματα του συστήματος επικοινωνίας που υλοποιήσαμε. Αντίστοιχα, στα σχήματα 59, 60 και 61 φαίνονται στιγμιότυπα από την προσομοίωση και την επαλήθευση του ανώτατου ιεραρχικού επιπέδου.



Σχήμα 59. Στιγμιότυπο από την προσομοίωση του ανώτατου ιεραρχικού επιπέδου με το ModelSim.

25	-2048	0	-8943	4578	-7884	-2750	-186	6161
26								
27	2231	9723	6473	-6564	3472	1112	1120	3864
28								
29	-2297	-2897	-3180	516	5849	-13666	706	-2354
30								
31	-3341	13213	2323	-7574	1016	-11394	3749	-3441
32								
33	4096	0	2164	-2874	-129	-2466	-5303	5533
34								
35	443	-6564	-82	-7426	-1386	-304	8892	-6861
36								
37	6392	-2897	734	-121	-2144	2888	-3964	8141
38								
39	664	6331	29	-510	1203	-5407	-4541	1751
40								
41	-9111	3290	268	915	714	-6571	-6357	-1334
42								
43	4071	5333	8099	9103	6832	-987	11333	-9162
44								
45	-2595	383	-8749	585	-105	-2837	-3099	6039
46								
47	-7145	5354	-1571	682	6323	1215	1252	-1480
48								
49	-5724	-3239	153	3809	-1380	14971	-6743	1495
50								
51	-1263	-11479	1168	8635	-4655	10510	-10040	-4059
52								
53	-7017	1827	3632	9176	4581	6729	-3641	1632
54								
55	-2803	106	1027	-7737	1673	-4670	2181	11614
56								
57	1295	-1401	2315	-5795	-533	6599	-2288	-632

Σχήμα 60. Στιγμιότυπο από το αρχείο που παράχθηκε κατά την προσομοίωση του ανώτατου ιεραρχικού επιπέδου.

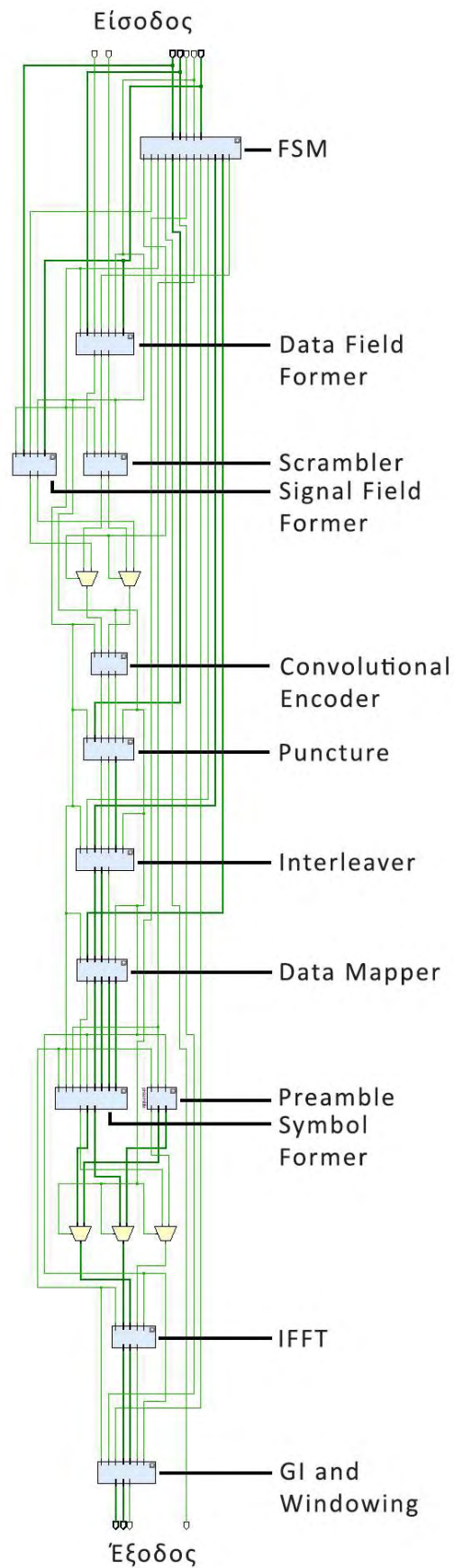

```

GI_and_windowing:
1890, -1672 || -3094, 5072 || -452, -136 || 3308, -1396 ||
3028, -2592 || -4014, -6469 || -7955, 518 || 892, 3265 ||
9496, 2207 || 35, -2990 || -3827, -7906 || 2590, 38 ||
-1870, 2661 || 127, -4293 || 980, -3549 || 657, -1873 ||
543, -7772 || -8787, 122 || 4190, 5200 || 6249, -6667 ||
-4528, -913 || 10202, 2422 || 3075, -524 || -4976, 1619 ||
7665, -9377 || 3652, -2750 || 146, 4902 || -2562, -3782 ||
-6016, 948 || -2710, 3051 || -3812, 6045 || 780, 10063 ||
5181, 5934 || -4420, 1111 || -6707, -2089 || 2526, 5534 ||
-2334, 917 || -95, -3039 || 12770, 8611 || 2533, 4399 ||
-430, 2974 || 3365, 526 || -4853, -7111 || -2142, 4560 ||
-1859, 11522 || -2709, 2970 || 897, -5503 || 3554, -2633 ||
7229, -1295 || 904, -1367 || 378, 9088 || 510, 698 ||
-3937, -2594 || 498, 11721 || 509, 1312 || 2890, -7451 ||
1402, -985 || -499, -3387 || 5944, -7130 || -1636, -2648 ||
-3208, 412 || -2817, -2701 || -11646, -1729 || || -4814, -3727 ||
0, -2048 || -3094, 5072 || -452, -136 || 3308, -1396 ||
3028, -2592 || -4014, -6469 || -7955, 518 || 892, 3265 ||
9496, 2207 || 35, -2990 || -3827, -7906 || 2590, 38 ||
-1870, 2661 || 127, -4293 || 980, -3549 || 657, -1873 ||

```

Σχήμα 61. Στιγμιότυπο από την επιβεβαίωση της σωστής λειτουργίας του ανώτατου ιεραρχικού επιπέδου με την βοήθεια προγράμματος σε C.

Τέλος, εφόσον έχουν ολοκληρωθεί οι σχεδιάσεις όλων των κυκλωμάτων, το συνολικό σύστημα θα περαστεί από σύνθεση. Με την σύνθεση θα εξακριβωθεί το κόστος του συστήματος σε υπολογιστικούς πόρους του FPGA. Τα αποτελέσματα της επιτυχούς σύνθεσης φαίνονται αναλυτικά στο κεφάλαιο 3.3, όπου δίνεται η συνολική κατανάλωση πόρων με στιγμιότυπα από το εργαλείο Vivado της Xilinx. Επιπλέον, διαπιστώνεται και η ομοιότητα του σχηματικού διαγράμματος της σχεδίασης που υλοποιήθηκε με το σχηματικό διάγραμμα που παρήγαγε το εργαλείο Vivado μετά την σύνθεση. Το σχηματικό διάγραμμα που υλοποιήθηκε είναι στο σχήμα 44 της ενότητας 4.1, ενώ το σχηματικό διάγραμμα του Vivado είναι στο σχήμα 62.



Σχήμα 62. Στιγμιότυπο από το σχηματικό διάγραμμα που παράχθηκε από το Vivado.

6 Συμπεράσματα

Όσον αφορά την υλοποίηση του πομπού, έχει σχεδιαστεί με τρόπο που είναι ανεξάρτητος τεχνολογίας. Άρα μπορεί να υλοποιηθεί σε οποιοδήποτε FPGA διαθέτει αρκετούς πόρους ώστε να χωράει η σχεδίαση. Για να επιτευχθεί αυτή η ανεξαρτησία, έχει ακολουθηθεί τεχνική διατήρησης ενός μικρού κρίσιμου μονοπατιού. Όπως έχει προαναφερθεί, το κρίσιμο μονοπάτι τη σχεδίασης αποτελείται από έναν πολλαπλασιαστή και ένα προσθήτη στην σειρά. Μία άλλη παρατήρηση για την υλοποίηση της σχεδίασης είναι ότι ο Interleaver έχει υλοποιηθεί με post defined μέθοδο, ώστε να χρησιμοποιηθεί λιγότερο πολύπλοκη λογική και να μειωθούν οι δεσμευόμενοι πόροι της σχεδίασης. Για να γίνει αυτό, χρησιμοποιήθηκαν μνήμες που περιέχουν τις διευθύνσεις που χρειάζονται, οπότε σε άλλη περίπτωση θα χρειαζόταν να υλοποιηθούν οι πολύπλοκες μαθηματικές σχέσεις που τις παράγουν. Επιπλέον, υπάρχει ευελιξία στην τοποθέτηση των πιλοτικών σημάτων του συμβόλου, έτσι η ανίχνευση του πακέτου μπορεί να γίνεται αποδοτικότερα σε περιβάλλοντα με μεγάλη ποικιλία θορύβων. Η σχεδίαση όταν λαμβάνει το πακέτο δεδομένων που παρέχει το κεφάλαιο Annex L του αρχείου του πρωτοκόλλου, εξάγει και την τελευταία πληροφορία μετά από 1041 κύκλους ρολογιού. Το πακέτο εισόδου έχει μήκος 800 bit. Έτσι για υψηλές συχνότητες του ρολογιού μπορεί να επιτευχθεί throughput της τάξης των Gigabits.

Η σχεδίαση του φυσικού επιπέδου του πομπού του πρωτοκόλλου IEEE 802.11a μπορεί να χρησιμοποιηθεί ως βάση και για τα πρωτόκολλα IEEE 802.11n και ac, τα οποία χρησιμοποιούνται σε εφαρμογές Internet Of Things (IOT). Επιπρόσθετα, αποτελεί βάση και για τα πρωτόκολλα IEEE 802.11p (v2n και v2x), τα οποία χρησιμοποιούνται για επικοινωνία μεταξύ οχημάτων και οποιουδήποτε άλλου σταθμού. Τα πρωτόκολλα IEEE 802.11p χρησιμοποιούνται πολύ στην σύγχρονη βιομηχανία.

Η ολοκλήρωση της εργασίας αυτής προσέφερε μία γενική εικόνα γύρο από το φυσικό επίπεδο των τηλεπικοινωνιακών συστημάτων. Μας έδωσε να καταλάβουμε το πλήθος των προβλημάτων και των απαιτήσεων που ένας σχεδιαστής πρωτοκόλλων επικοινωνίας καλείται να λάβει υπόψιν του. Μας έδειξε το μέγεθος της πολυπλοκότητας των ασύρματων τηλεπικοινωνιακών συστημάτων. Μας προσέφερε γνώση σε βάθος γύρο από το φυσικό επίπεδο του πομπού του πρωτοκόλλου επικοινωνίας IEEE 802.11a. Τέλος, μας βοήθησε να εμβαθύνουμε τις προγραμματίσιμες μας ικανότητες με την γλώσσα περιγραφής υλικού VHDL και να κατανοήσουμε τις σχεδιάσιμες δυσκολίες που μπορεί να προκύψουν.

7 Μελλοντικές εργασίες

Στο μέλλον ο σκοπός της εργασίας μπορεί να επεκταθεί. Δύο σημαντικές αναβαθμίσεις είναι η υλοποίηση του δέκτη του συστήματος επικοινωνίας και η τροποποίηση των σχεδιάσεων ώστε να μπορούν να τυπωθούν σε ASIC κυκλώματα. Η υλοποίηση του δέκτη θα χρειαζόταν επιπλέον σχεδίαση για την υλοποίηση σημαντικών κυκλωμάτων, όπως το σύστημα της ανίχνευσης του πακέτου δεδομένων που στάλθηκε από τον πομπό, το σύστημα του ευθύ γρήγορου μετασχηματισμού Fourier και τον αποκωδικοποιητή Viterbi [9]. Ενώ θα χρειαζόταν και η τροποποίηση των υπόλοιπων κυκλωμάτων ώστε να μπορούν να αναστρέψουν την επεξεργασία που έκαναν στο πακέτο δεδομένων. Επιπρόσθετα, θα μπορούσε να σχεδιαστούν συστήματα που αποτελούνται και από πομπό και από δέκτη, ώστε να επιτευχθεί επικοινωνία και προς τις δύο κατευθύνσεις. Αυτά τα συστήματα ονομάζονται πομποδέκτες. Αυτό θα βοηθούσε στην επέκταση των γνώσεων γύρο από τα συστήματα επικοινωνίας και από τις δύο πλευρές, του πομπού και του δέκτη. Ενώ ως δεύτερο βήμα, για την εκμάθηση σε βάθος μιας πιο μεθοδικής σχεδίασης, το σύστημα πομποδέκτη θα μπορούσε να τροποποιηθεί ώστε να υλοποιηθεί ως ASIC κύκλωμα.

8 Παραρτήματα

8.1 Αναπαράσταση ως προς 2 αριθμών με σταθερή υποδιαστολή

Στα κυκλώματα που υλοποιήθηκαν κάθε πληροφορία αριθμών με σταθερή υποδιαστολή αναπαρίσταται με τρόπο που απλοποιεί τις αριθμητικές πράξεις μεταξύ αυτών. Συγκεκριμένα, κάθε αριθμός σταθερής υποδιαστολής πολλαπλασιάζεται με τον αριθμό 65536 και το ακέραιο τμήμα από το αποτέλεσμα που προκύπτει είναι ο αριθμός που θα χρησιμοποιηθεί για αν γίνουν οι ζητούμενες πράξεις. Ο πολλαπλασιασμός αυτός είναι νοητός, δηλαδή δεν εκτελείται πραγματικά κάποια πράξη ώστε να υπολογιστεί ο ακέραιος αριθμός που αντιστοιχεί στο αριθμό με σταθερή υποδιαστολή. Όμως επειδή οι αριθμοί σταθερής υποδιαστολής, που αρχικά θα χρειαστούν, είναι προκαθορισμένοι από το σύστημα διαμόρφωσης, αντί να αποθηκευτούν στους καταχωρητές όπως είναι, αποθηκεύονται ως ακέραιοι αριθμοί με την αλλαγή που ορίστηκε.

Για παράδειγμα ο αριθμός 0.707106 θα πολλαπλασιαστεί νοητά με τον αριθμό 65536 και το ακέραιο τμήμα από το αποτέλεσμα που θα προκύψει θα αποθηκευτεί σε ένα από τους καταχωρητές του συστήματος διαμόρφωσης, ώστε να τον χρησιμοποιήσει όποτε τον χρειαστεί. Το αποτέλεσμα της παραπάνω πράξης φαίνεται στην σχέση 18, όπου από αυτό θα κρατηθεί μόνο το ακέραιο μέρος.

Σχέση 18

$$0.707106 * 65536 = 46340.898816$$

Στην συνέχεια οι τιμές αυτές θα σταλθούν στα επόμενα κυκλώματα και οι πράξεις που θα γίνει θα έχουν μικρότερη πολυπλοκότητα από τις αντίστοιχες πράξεις μεταξύ αριθμών σταθερής υποδιαστολής. Στο τέλος, μετά την εκτέλεση οποιασδήποτε πράξης ο ακέραιος αριθμός μπορεί να διαιρεθεί πάλι με τον αριθμό 65536, ώστε να ανακτηθεί ο ζητούμενος αριθμός σταθερής υποδιαστολής. Όπως φαίνεται και στην σχέση 19, ο αριθμός που προκύπτει μπορεί να μην είναι ακριβώς ο ίδιος με τον αρχικό, αλλά η ακρίβεια του είναι αρκετή ώστε η τελική πληροφορία να ανακτηθεί πλήρως.

Σχέση 19

$$46340 / 65536 = 0.70709$$

8.2 Αναπαράσταση μιγαδικών αριθμών

Ένας μιγαδικός αριθμός αποτελείται από δύο τμήματα, το πραγματικό και το φανταστικό μέρος. Στην σχέση 20, ο μιγαδικός αριθμός z αποτελείται από το πραγματικό μέρος x και το φανταστικό μέρος y .

Σχέση 20

$$z = x + yi$$

Στην υλοποίηση των κυκλωμάτων τα τμήματα ενός μιγαδικού αριθμού αποθηκεύονται σε δύο διαφορετικές θέσεις μνήμης. Στην συνέχεια το κάθε ένα από αυτά χειρίζεται με τον τρόπο που ορίζουν οι πράξεις στο πεδίο των μιγαδικών αριθμών. Μπορεί να τύχει ένα από τα δύο τμήματα του μιγαδικού αριθμού να είναι μηδενικό, όπως φαίνεται στις σχέσεις 21 και 22.

Σχέση 21

$$z_1 = 0 + yi = yi$$

Σχέση 22

$$z_2 = x + 0i = x$$

Σε αυτή την περίπτωση η αντίστοιχη θέση μνήμης θα περιέχει και αυτή μηδενική τιμή. Έτσι δεν θα υπάρχει αλλαγή στην εκτέλεση των αριθμητικών πράξεων. Στην υλοποίηση το φανταστικό και το πραγματικό τμήμα θα είναι αριθμοί σταθερής υποδιαστολής.

8.3 Πράξεις με μιγαδικούς αριθμούς και ικανοποίηση του ζητούμενου κρίσιμου μονοπατιού

Η μεγαλύτερη σειρά πράξεων χρησιμοποιείται από το σύστημα IFFT. Συγκεκριμένα η πράξη αυτή χρησιμοποιεί δύο προσθέσεις και ένα πολλαπλασιασμό. Η σειρά των πράξεων αυτών προκύπτει από το σύστημα butterfly του IFFT επειδή θα γίνουν μέσα στο σύνολο των μιγαδικών αριθμών. Η δομή butterfly στην χειρότερη περίπτωση προσθέτει δύο μιγαδικούς αριθμούς και στην συνέχεια το αποτέλεσμα το πολλαπλασιάζει με έναν ακόμη μιγαδικό αριθμό. Στις σχέσεις 23 και 24 φαίνονται οι πράξεις πρόσθεσης και πολλαπλασιασμού, αντίστοιχα, μεταξύ δύο μιγαδικών αριθμών. Ενώ στην σχέση 25 φαίνεται ο βασικός κανόνας των μιγαδικών αριθμών.

Σχέση 23

$$i^2 = -1$$

Σχέση 24

$$(x + yi) + (a + bi) = (x + a) + (y + b)i$$

Σχέση 25

$$z_1 * z_2 = (x_1 * x_2 - y_1 * y_2) + (x_1 y_2 + x_2 y_1)i$$

Στη δομή butterfly της χειρότερης περίπτωσης, ο z_1 της σχέσης 25 αποτελείται από μία πρόσθεση δύο προηγούμενων μιγαδικών αριθμών. Έτσι, για να αναδειχθεί το πρόβλημα, αν ενωθούν οι σχέσεις 24 και 25 θα ληφθεί η σχέση 26.

Σχέση 26

$$a = k + li$$

$$b = m + ni$$

$$z_1 = a + b = (k + m) + (l + n)i$$

$$z_1 * z_2 = ((k + m) * x_2 - (l + n) * y_2) + ((k + m) y_2 + x_2 (l + n))i$$

Όπως φαίνεται για τον υπολογισμό του πραγματικού ή του φανταστικού αριθμού απαιτούνται μία πρόσθεση για τα $(k + m)$ και $(l + n)$ παράλληλα και ακολουθούν στην σειρά ο πολλαπλασιασμός και μία πρόσθεση ή αφαίρεση ακόμα. Αυτή η ακολουθία πράξεων ξεπερνάει το ζητούμενο κρίσιμο μονοπάτι, το οποίο έχει προκαθοριστεί ως μία πράξη πρόσθεσης και μία πολλαπλασιασμού. Για αυτό τον λόγο στο σύστημα του IFFT έχουν σπάσει τα μονοπάτια που ξεπερνούν το ζητούμενο κρίσιμο μονοπάτι. Έτσι έχουν αυξηθεί τα στάδια του IFFT αλλά έχει διατηρηθεί το μήκος του κρίσιμου μονοπατιού.

9 Αναφορές

- [1] IEEE Std 802.11 TM, «Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,» 2012.
- [2] H. Zimmermann, «OSI Reference Model - The OSI Model of Architecture for Open Systems Interconnection,» 1980.
- [3] N. K. Randive και G. P. Prof Brokhade, «Design and Implementation of Data Scrambler & Descrambler System using VHDL,» Global Journals Inc, 2015.
- [4] S. Haykin και M. Moher, Συστήματα Επικοινωνίας, Αθήνα: Εκδόσεις Παπασωτηρίου, 2010.
- [5] Σ. Θεοδωρίδης, Κ. Μπερμπερίδης και Λ. Κοφίδης, Εισαγωγή στη θεωρία σημάτων και συστημάτων, Αθήνα: τυπωθήτω Γιώργος Δαρδάνος, 2005.
- [6] N. Andgart, «Peak and Power Reduction in Multicarrier Communication Systems,» 2005.
- [7] Xilinx Inc, «ise_c_fpga_design_flow_overview,» 2008. [Ηλεκτρονικό]. Available: https://www.xilinx.com/itp/xilinx10/isehelp/ise_c_fpga_design_flow_overview.htm.
- [8] V. Loncke.Ευρεσιτεχνία US 12/413,511, 2013.
- [9] M. S. Ryan και G. R. Nudd, The Viterbi Algorithm, Department of Computer Science, University of Warwick, Coventry, CV4 7AL, England, 1993