



ΕΘΝΙΚΟ ΜΕΤΣΟΒΕΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ
ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΨΗΦΙΑΚΟΥ ΠΟΜΠΟΔΕΚΤΗ ΟFDM ΣΕ FPGA

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικόλαος Ι. Γκίκας
Βασίλειος Α. Γουβάλας

Επιβλέποντες

Κωνσταντίνος Μαλιάτσος
Υποψήφιος Διδάκτωρ ΕΜΠ

Γιώργος Οικονομάκος
Επίκουρος Καθηγητής ΕΜΠ

ΑΘΗΝΑ

ΙΟΥΛΙΟΣ 2011

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΨΗΦΙΑΚΟΥ ΠΟΜΠΟΔΕΚΤΗ OFDM ΣΕ FPGA

ΝΙΚΟΛΑΟΣ Ι. ΓΚΙΚΑΣ
ΒΑΣΙΛΕΙΟΣ Α. ΓΟΥΒΑΛΑΣ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟ-
ΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ
ΕΘΝΙΚΟ ΜΕΤΣΟΒΕΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

DIPLOMA THESIS
DESIGN AND IMPLEMENTATION OF A
DIGITAL OFDM TRANCEIVER IN FPGA
NIKOLAOS I. GKIKAS
VASILEIOS A. GOUVALAS
SCHOOL OF ELECTRICAL AND
COMPUTER ENGINEERING
NATIONAL TECHNICAL UNIVERSITY OF ATHENS

Επιβλέποντες

Κωνσταντίνος Μαλιάτσος
Υποψήφιος Διδάκτωρ ΕΜΠ

Γιώργος Οικονομάκος
Επίκουρος Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 19^η Ιουλίου 2011.

Φίλιππος Κωνσταντίνου
Καθηγητής ΕΜΠ

Θανάσης Παναγόπουλος
Λέκτορας ΕΜΠ

Γιώργος Οικονομάκος
Επίκουρος Καθηγητής ΕΜΠ

Νικόλαος Ι. Γκίκας
Διπλωματούχος Ηλεκτρολόγος Μηχανικός & Μηχανικός Υπολογιστών ΕΜΠ

Βασίλειος Α. Γουβάλας
Διπλωματούχος Ηλεκτρολόγος Μηχανικός & Μηχανικός Υπολογιστών ΕΜΠ

©2011 *Νικόλαος Γκίκας, Βασίλειος Γουβάλας*
Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται αυστηρά η αντιγραφή, αποθήκευση και διανομή με οποιοδήποτε μέσο, της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικούς σκοπούς.

Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή της παρούσας εργασίας, για σκοπό μη κερδοσκοπικό και μη εμπορικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο, εκφράζουν τους συγγραφείς και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

© 2011 *Nikolaos Gkikas, Vasileios Gouvalas*
All rights reserved.

It is strictly prohibited to copy, store and distribute by any means, this work, in whole or in part, for commercial purposes.

It is allowed to reprint, store and distribute this work for non-profitable, non-commercial purposes, to assist education or research, with the requirement to refer to the source and to preserve the current message. Questions related to commercial usage should be asked to the authors.

The opinions and conclusions contained in this document, express only the authors and it should not be implied that they represent the official opinions of the National Technical University of Athens.

Περίληψη

Στα πλαίσια της εργασίας αυτής, κατασκευάστηκε κύκλωμα FPGA που υλοποιεί ένα πλήρες τηλεπικοινωνιακό σύστημα OFDM (πομποδέκτης). Η περιγραφή του κυκλώματος έγινε με τη γλώσσα VHDL.

Τα τεχνικά του χαρακτηριστικά περιλαμβάνουν : Real-Time λειτουργία σε simplex τρόπο μετάδοσης (Simplex TX ή Simplex RX). Πλήρως επεκτάσιμη αρχιτεκτονική με κατανομή των τηλεπικοινωνιακών λειτουργιών σε κυκλωματικές βαθμίδες (modules). Ανάπτυξη εσωτερικής διεπαφής επικοινωνίας μεταξύ των βαθμίδων. Δυνατότητα καθορισμού πλήθους subcarrier (μεγέθους FFT) και μήκους του κυκλικού προθέματος. Δυνατότητα επιλογής ξεχωριστής διαμόρφωσης ανά subcarrier (QAM4, QAM16, QAM64). Παραμετροποιήσιμη αρχιτεκτονική με υλοποίηση συστήματος διανομής και συγχρονισμού παραμέτρων.

Η αλυσίδα πομπού περιλαμβάνει τα στάδια : χειρισμός δεδομένων εισόδου, σειριακό-σε-παράλληλο, διαμόρφωση κατά QAM, εισαγωγή training συμβόλων, μετασχηματισμός IFFT (διαμόρφωση κατά OFDM), εισαγωγή κυκλικού προθέματος.

Στην αλυσίδα του δέκτη δόθηκε μεγάλη έμφαση στα στάδια του συγχρονισμού στο χρόνο και στη συχνότητα (αλγόριθμος Schmidl & Cox / Bhargava), καθώς και στη διαδικασία της αριθμητικής προσέγγισης στοιχειωδών συναρτήσεων : Μέθοδος του S. Rajan κ.α για προσέγγιση της \arctan , Μέθοδος του Tang για προσέγγιση των \cos/\sin (που χρησιμοποιούνται στα στάδια της εκτίμησης και αντιστάθμισης ολίσθησης συχνότητας, αντίστοιχα). Τα υπόλοιπα στάδια της αλυσίδας του δέκτη είναι : αφαίρεση κυκλικού προθέματος, μετασχηματισμός FFT (αποδιαμόρφωση κατά OFDM), εκτίμηση καναλιού (αλγόριθμος LS), αποδιαμόρφωση κατά QAM, παράλληλο-σε-σειριακό, χειρισμός δεδομένων εξόδου.

Υλοποίηση πλήρους μοντέλου του συστήματος OFDM σε περιβάλλον MATLAB για το λειτουργικό και αριθμητικό έλεγχο του κυκλώματος.

Διενέργεια μαζικών προσομοιώσεων μετάδοσης μέσα από στοχαστικά μοντέλα καναλιών (AWGN, Multipath), για την τηλεπικοινωνιακή αποτίμηση του συστήματος.

Abstract

In the context of this work, an FPGA circuit was constructed, implementing a full telecommunication OFDM system (transceiver). The circuit description was made with the VHDL language.

The key technical features include: Real-Time operation in simplex transmission mode (Simplex TX, Simplex RX). Fully expandable architecture, with the implementation of the various telecommunication functions as discrete circuit modules. Adjustable number of subcarriers (and accordingly FFT size) and length of cyclic prefix. Capability to select different modulation schema per subcarrier (QAM4, QAM16, QAM64). Parametrized architecture with the implementation of a parameter distribution and synchronization system.

The transmitter chain stages are: input data manipulation, serial-to-parallel, QAM modulation, training symbols insertion, IFFT transform (OFDM modulation), cyclic prefix insertion.

In the receiver chain, great emphasis was given to the timing and frequency synchronization stages (Schmidl & Cox / Bhargava algorithm), as well as to the realization of arithmetic approximation of elementary functions: S. Rajan's method for arctan approximation and Tang's method for cos/sin approximation (that were used for the frequency offset estimation and compensation stages, correspondingly). The rest of the receiver stages are: cyclic prefix removal, FFT transform (OFDM demodulation), channel estimation (LS algorithm), QAM demodulation, parallel-to-serial, output data manipulation.

Implementation of the entire OFDM system as a MATLAB model, for the functional and arithmetical evaluation of the circuit.

Massive transmission simulations, through stochastic channel models (AWGN, Multipath), for the evaluation of the telecommunication performance of the system.

Περιεχόμενα

1	Εισαγωγικά	17
1.1	Περί Ψηφιακών Τηλεπικοινωνιών	17
1.1.1	Πομπός	19
1.1.2	Τύποι διαμορφώσεων	19
1.1.3	Δέκτης	20
1.1.4	Κανάλι	21
1.2	Περί Ψηφιακών Συστημάτων	22
1.2.1	Απο τη Θεωρία στην Πράξη	22
1.2.2	Ψηφιακά Συστήματα - Σχεδίαση	24
1.2.3	Δομικά Στοιχεία Συστημάτων CAD - Γλώσσες Περιγραφής Υλικού - Μέρος Α	27
1.2.4	Τεχνολογίες Υλοποίησης	30
1.2.5	Δομικά Στοιχεία Συστημάτων CAD - Μέρος Β	33
1.2.6	Ένα Διαφωτιστικό Παράδειγμα	36
2	Τηλεπικοινωνιακές προδιαγραφές	41
2.1	Μέρος Α - Θεωρητική ανάλυση συστήματος OFDM	41
2.1.1	Γενικά	41
2.1.2	Αρχές λειτουργίας	41
2.1.3	Περιγραφή λειτουργίας	43
2.1.4	Επεξηγήσεις	43
2.1.5	Μοντέλο	46
2.1.6	Κανάλι - Θόρυβος	48
2.1.7	Παράμετροι OFDM	52
2.2	Μέρος Β - Πρακτική Υλοποίηση Συστήματος	53
2.2.1	Παραδοχές	53
2.2.2	Παράμετροι	56
2.2.3	Συγχρονισμός στο Χρόνο	56
2.2.4	Συγχρονισμός στη Συχνότητα	62

2.2.5	Εκτίμηση Καναλιού	64
3	Το Σύστημα <i>per se</i>	67
3.1	Εξωτερική Όψη του Συστήματος	67
3.1.1	Βασικά Χαρακτηριστικά	67
3.1.2	Περιγραφή Ακροδεκτών Εισόδου/Εξόδου	70
3.1.3	Παραμετροποίηση Συστήματος	73
3.2	Πλήρης Αλυσίδα Συστήματος Πομποδέκτη	77
3.2.1	Διεπαφή Επικοινωνίας Υπομονάδων	79
3.2.2	Ελεγκτής Τρόπου Μετάδοσης	84
3.2.3	Το module <i>Altera OFDM Core</i>	85
3.2.4	Σχολιασμός της μετα-FFT Κλιμάκωσης (<i>Post-FFT Scaling</i>)	88
3.2.5	Σχολιασμός της προ-FFT Κλιμάκωσης (<i>Pre FFT scaling</i>)	89
3.3	Δομικές Μονάδες Μέτρησης της Πληροφορίας	90
3.4	Αλυσίδα Υποσυστήματος Πομπού	91
3.4.1	Η Διαδρομή Δεδομένων στον Πομπό (<i>TX Datapath</i>)	91
3.4.2	Συστημική Περιγραφή Μονάδων	93
3.5	Αλυσίδα Υποσυστήματος Δέκτη	99
3.5.1	Η Διαδρομή Δεδομένων στο Δέκτη (<i>RX Datapath</i>)	99
3.5.2	Συστημική Περιγραφή Μονάδων	101
4	Μέσα στα μύρα κουτιά	115
4.1	Εισαγωγικά	115
	TX modules	115
4.2	Disassembler [DSA]	115
4.2.1	Η Top-Level Όψη	115
4.2.2	Περιγραφή Κυκλωματικής Λειτουργίας	117
4.2.3	Τα Επιμέρους Υποσυστήματα	119
4.3	Symbol Mapper [SMA]	122
4.3.1	Η Top-Level Όψη	122
4.3.2	Περιγραφή Κυκλωματικής Λειτουργίας	122
4.3.3	Τα Επιμέρους Υποσυστήματα	124
4.4	Frame Generator [FRG]	125
4.4.1	Η Top-Level Όψη	125
4.4.2	Περιγραφή Κυκλωματικής Λειτουργίας	125
4.4.3	Τα Επιμέρους Υποσυστήματα	128

RX modules	128
4.5 Αριθμητική	128
4.5.1 Signed/Unsigned Integer	129
4.5.2 Fixed Point	130
4.5.3 Στρογγυλοποίηση	131
4.5.4 Υπερχείλιση - Κορεσμός	131
4.6 Time Synchroniser [TSY]	132
4.6.1 Η Top-Level Όψη	132
Τα Επιμέρους Υποσυστήματα	134
4.6.2 Κυκλικοί Buffer [TSY.CBW_*]	134
4.6.3 Metric Computer [TSY.DCR]	137
4.6.4 Decision Unit [TSY.DCU]	140
4.6.5 Controller [TSY.CO]	143
4.6.6 ARG Computer [TSY.ARG]	147
4.7 Frequency Compensator [FCO]	153
4.7.1 Υπολογιστής Ορίσματος - Omega engine	154
4.7.2 Υπολογιστής cos/sin - Cos engine	156
4.7.3 Αντιστάθμιση - Compensation engine	164
4.8 Fine Frequency Estimator [FFE]	164
4.8.1 Η Top-Level Όψη	164
4.8.2 Περιγραφή Κυκλωματικής Λειτουργίας	164
4.9 Channel Estimator/Equalizer [CEQ]	166
4.9.1 Η Top-Level Όψη	167
Τα Επιμέρους Υποσυστήματα	170
4.9.2 Μνήμη Συντελεστών [CEQ.COEFF_RAM]	170
4.9.3 Μνήμη Συμβόλων [CEQ.SYMBOL_ROM]	171
4.9.4 Μηχανή Εκτίμησης [CEQ.ESTIMATION_ENGINE]	171
4.9.5 Μηχανή Ισοστάθμισης [CEQ.COMPENSATION_ENGINE]	175
4.10 Symbol Demapper [DMP]	175
4.10.1 Η Top-Level Όψη	176
Τα Επιμέρους Υποσυστήματα	176
4.10.2 RX Constellation Controller [DMP.RCC]	176
4.10.3 Deconstellator [DMP.DCO]	177
4.11 Assembler [ASM]	181
4.11.1 Η Top-Level Όψη	181

4.11.2 Περιγραφή Κυκλωματικής Λειτουργίας	181
5 Αποτίμηση Συστήματος	189
5.1 Λειτουργικός Έλεγχος Συστήματος	189
5.1.1 Το Testbench	189
5.1.2 Έλεγχος Μονάδας (Unit Testing)	190
5.1.3 Έλεγχος Πλήρους Συστήματος (System Testing)	191
5.1.4 Πομπός	191
5.1.5 Δέκτης	194
5.2 Τηλεπικοινωνιακή Αποτίμηση Δέκτη	196
5.2.1 Κανάλι	196
5.2.2 Script Αποτίμησης (Evaluation Scripts)	197
5.2.3 Πορίσματα Αποτίμησης	198
Βιβλιογραφία	207

Κατάλογος σχημάτων

1.1	Σχεδιάγραμμα ψηφιακού τηλεπικοινωνιακού συστήματος	18
1.2	Γενική Διαδικασία Σχεδίασης Προϊόντων	25
1.3	Σχεδίαση Βαθμίδος	26
1.4	Κύκλος Ανάπτυξης Ψηφιακού Συστήματος	27
1.5	Χρήση του Graphical Editor (Altera Quartus II CAD)	28
1.6	Α' Στάδιο Σχεδίασης CAD	30
1.7	Γενική Δομή του FPGA	31
1.8	Λεπτομέρεια Δομής FPGA	32
1.9	Πίνακας LUT	32
1.10	Παραδείγματα Λογικών Βαθμίδων Εμπορικών Συστημάτων FPGA	34
1.11	Διασύνδεση JTAG	34
1.12	Β' Στάδιο Σχεδίασης CAD	36
1.13	Αρχικό Λογικό Κύκλωμα	37
1.14	Βελτιστοποιημένο Λογικό Κύκλωμα	38
1.15	Κυματομορφές Λειτουργικής Προσομοίωσης	39
1.16	Προγραμματισμένοι LUT	40
1.17	Τμήμα του προγραμματισμένου FPGA	40
2.1	Φάσμα subcarrier OFDM	42
2.2	Τηλεπικοινωνιακό μπλοκ διάγραμμα πομπού και δέκτη OFDM	44
2.3	Φάσμα εκπεμπόμενου OFDM σήματος με raised cosine pulse shaping	46
2.4	Λειτουργικό διάγραμμα του υλοποιηθέντος συστήματος OFDM	54
2.5	Τετραγωνικοί αστερισμοί QAM	55
2.6	TDD frame	55
2.7	Περιθώριο σφάλματος συγχρονισμού	58
2.8	OFDM frame	58
2.9	Μετρικές συγχρονισμού: S&C και Μέθοδος A	60
2.10	Ολίσθηση συχνότητας subcarrier	63

2.11	Τύποι αλγορίθμων εκτίμησης καναλιού	65
2.12	Σύγκριση LS, MMSE αλγορίθμων εκτίμησης καναλιού	65
3.1	Το μοντέλο OSI	68
3.2	Διάφορες τιμές για το duty cycle	69
3.3	RXTXTOP - Chip I/O Pin Diagram	71
3.4	Συνοπτικό RXTXTOP Datapath	78
3.5	Τυπικά σήματα του Avalon ST Interface	81
3.6	Παράδειγμα μεταφοράς δεδομένων χωρίς backpressure	82
3.7	Παράδειγμα μεταφοράς παρουσία backpressure ($readyLatency = 4$)	83
3.8	Παράδειγμα μεταφοράς παρουσία backpressure ($readyLatency = 0$)	83
3.9	Διάγραμμα Ακροδεκτών TDD Controller	84
3.10	Διάγραμμα Ακροδεκτών Altera OFDM Core	85
3.11	Βασική αρχή λειτουργίας του OFDM Core	86
3.12	Κοινοχρησία του module OFDM Core	86
3.13	Συνεπτυγμένη αλυσίδα πομπού	90
3.14	Συνεπτυγμένη αλυσίδα δέκτη	91
3.15	Το TX Datapath	92
3.16	Διάγραμμα ακροδεκτών Disassembler	93
3.17	Διάγραμμα ακροδεκτών Symbol Mapper	95
3.18	Διάγραμμα ακροδεκτών Frame Generator	97
3.19	Εποπτεία λειτουργίας του module OFDM Core (TX Mode)	98
3.20	Το RX Datapath	100
3.21	Διάγραμμα ακροδεκτών Time Synchronizer	101
3.22	Διάγραμμα ακροδεκτών Frequency Compensator	104
3.23	Εποπτεία λειτουργίας του module OFDM Core (RX Mode)	105
3.24	Διάγραμμα ακροδεκτών Fine Frequency Estimator	106
3.25	Διάγραμμα ακροδεκτών Channel Estimator	108
3.26	Διάγραμμα ακροδεκτών Symbol Demapper	110
3.27	Διάγραμμα ακροδεκτών Assembler	112
4.1	DSA Top-Level διάγραμμα	116
4.2	MAP Top-Level διάγραμμα	122
4.3	FRG Top-Level διάγραμμα	126
4.4	TSY Top-Level διάγραμμα	133
4.5	TSY.CBW_* Top-Level διάγραμμα	135

4.6	TSY.DCR datapath και ακρίβειες εργασίας	141
4.7	TSY.CO Top-Level διάγραμμα	144
4.8	TSY.CO διάγραμμα καταστάσεων	145
4.9	Γράφημα της \arctan στο διάστημα $[-4,4]$	147
4.10	Σφάλμα προσέγγισης α' τάξης (E_1) της \arctan	148
4.11	Σφάλμα προσέγγισης γ' τάξης (E_2) της \arctan	149
4.12	Αρίθμηση τεταρτημορίων για την ανακατασκευή της \arctan	150
4.13	TSY.ARG datapath και ακρίβειες εργασίας	152
4.14	Σφάλμα του ARG computer	153
4.15	FCO Top-Level διάγραμμα	154
4.16	Υπολογιστικές μηχανές του FCO	158
4.17	Αρίθμηση τεταρτημορίων για το \cos engine	159
4.18	Κατάτμηση της Ω_c σε υπολέξεις	160
4.19	Κατάτμηση της x σε υπολέξεις	160
4.20	Σφάλμα του \cos engine για έναν πλήρη κύκλο της Ω_c	164
4.21	FFE Top-Level διάγραμμα	165
4.22	CEQ Top-Level διάγραμμα	168
4.23	Διάγραμμα καταστάσεων (CEQ)	170
4.24	Μηχανές υπολογισμού του CEQ	173
4.25	Σφάλμα υπολογισμών κατά την εκτίμηση ενός τυπικού καναλιού	174
4.26	Σύγκριση εκτίμησης καναλιού μεταξύ μοντέλου και θεωρίας, παρουσία AWGN	174
4.27	DMP Top-Level διάγραμμα	176
4.28	DMP.RCC Top-Level διάγραμμα	176
4.29	DMP.DCO Top-Level διάγραμμα	179
4.30	Κωδικοποίηση τεταρτημορίων για τα στάδια σύγκρισης (DMP.DCO)	180
4.31	ASM Top-Level διάγραμμα	182
4.32	Η άτυπη μηχανή καταστάσεων του Assembler	184
5.1	Μέσο απόλυτο σφάλμα εκτίμησης SOF	199
5.2	Διακύμανση σφάλματος εκτίμησης SOF	199
5.3	Ποσοστά σφαλμάτων συγχρονισμού	200
5.4	Μέσο απόλυτο σφάλμα εκτίμησης ολίσθησης συχνότητας	201
5.5	Διακύμανση σφάλματος εκτίμησης ολίσθησης συχνότητας	202
5.6	ISNR και INOISE στην έξοδο του Channel Estimator	203
5.7	Σύγκριση INOISE μεταξύ AWGN και ISI καναλιού	204

5.8 Σύγκριση INOISE μεταξύ κυκλώματος και μοντέλου	204
5.9 SER στην έξοδο του δέκτη	206
5.10 BER στην έξοδο του δέκτη	206

Κατάλογος πινάκων

1.1	Σημαντικοί σταθμοί στην ιστορία των τηλεπικοινωνιών	18
1.2	Συνοπτική αναδρομή σε σημεία σταθμούς στην ιστορία της ηλεκτρονικής	23
2.1	Χαρακτηρισμοί multipath καναλιών	52
2.2	Παράμετροι OFDM του συστήματος που υλοποιήθηκε	56
2.3	Παράμετροι συγχρονισμού στο χρόνο	62
3.1	Κατάταξη real-time συστημάτων	68
3.2	Παράμετροι του Top-Level (RXTXTOP)	74
3.3	Κωδικοποίηση παραμέτρου DUPLEX_MODE	75
3.4	Διανεμόμενες (dispatched) T/Π παράμετροι	77
3.5	Ακρωνύμια υπομονάδων (modules)	79
3.6	Γλωσσάρι για το Avalon ST Interface	80
3.7	Ρόλοι σημάτων στο Avalon ST Interface	81
3.8	Modules και Front/Back Pressure	82
3.9	Παράμετροι του frame generator	98
3.10	Παράμετροι του time sync	103
3.11	Παράμετροι του frequency compensator	105
3.12	Παράμετροι του fine frequency estimator	108
3.13	Παράμετροι του channel estimator	109
3.14	Μορφή του block πληροφορίας ανά τύπο QAM	111
4.1	Κατανομή ωφέλιμων και πλεονάζοντων bit ανά τύπο διαμόρφωσης QAM	115
4.2	Τα σήματα rd και wr ως ρυθμιστές της λειτουργίας του c_buffer	120
4.3	Εκκίνηση και λειτουργία κυκλώματος	123
4.4	Σχηματισμός διεύθυνσης ROM ανά τύπο διαμόρφωσης QAM	125
4.5	Τεχνικά Χαρακτηριστικά Altera ROM (MAP)	125
4.6	Τεχνικά Χαρακτηριστικά Altera ROM (FRG)	128
4.7	Τεχνικά Χαρακτηριστικά Altera RAM (CBW)	136

4.8	Προσέγγιση της \arctan σε όλα τα τεταρτημόρια	150
4.9	Σχέσεις εύρεσης τεταρτημορίου	151
4.10	Σχέσεις ανακατασκευής \arctan και τιμές των K, L	151
4.11	Αναγωγικοί τύποι των $\cos(\psi), \sin(\psi)$ στο πρώτο τεταρτημόριο	159
4.12	Διαστήματα προσέγγισης των \cos/\sin	160
4.13	Συντελεστές πολυωνύμων 1 ^{ης} τάξης της $\cos(\frac{\pi}{2}x)$ στο διάστημα $[0, 1)$	161
4.14	Τελικοί συντελεστές \hat{C}_1, \hat{C}_0 και οι αντίστοιχοι stored integers	162
4.15	Τύποι ανακατασκευής των $g_c(\Omega_c), g_s(\Omega_c)$	163
4.16	Τεχνικά Χαρακτηριστικά Altera RAM (CEQ)	171
4.17	Τεχνικά Χαρακτηριστικά Altera ROM (CEQ)	171
4.18	Περιεχόμενα των LUT_RC16 και LUT_RC64	181
4.19	"Θύρες" Ε/Ε της διεργασίας FSM	184

Κεφάλαιο 1

Εισαγωγικά

1.1 Περί Ψηφιακών Τηλεπικοινωνιών

Ιστορικά

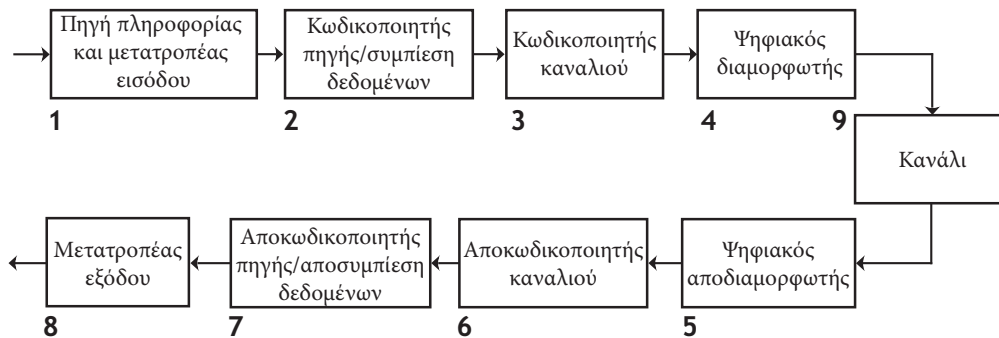
Ο δρόμος που έχει διανυθεί στην ιστορία των τηλεπικοινωνιών, ώστε να φτάσουμε στις σημερινές τεχνολογίες, είναι πολύ μακρύς και έχει τις απαρχές του στο 19^ο αιώνα με την εφεύρεση του ηλεκτρικού τηλέγραφου. Από τον τηλέγραφο, εκείνης της εποχής, μέχρι τις ψηφιακές τηλεπικοινωνιακές μηχανές του τέλους του 20ου αιώνα, πολλοί εφευρέτες και ερευνητικές ομάδες συνέβαλαν με το έργο τους. Η εξέλιξη κάθε άλλο παρά γραμμική ήταν, με την κάθε εποχή να προβάλλει τις αντιστάσεις της στην πορεία της "εξέλιξης". Τεχνολογικές δυσκολίες, διαμάχες μεταξύ εφευρετών, κανόνες που επέβαλαν η αγορά και τα διάφορα κράτη, είναι μερικοί από τους παράγοντες που καθόρισαν την πορεία της εξέλιξης. Ωστόσο ιδωμένες από μία *a-posteriori* ματιά οι μηχανές τηλεπικοινωνιών αποτελούν μία πυραμίδα τεχνικών, ανακαλύψεων και γνώσεων. Ο πίνακας 1.1 επιχειρεί μία συνοπτική παρουσίαση σημείων σταθμών στην ιστορία των επικοινωνιών.

1799	Alessandro Volta: Ηλεκτρικό στοιχείο
1837	Samuel Morse: Ηλεκτρικός τηλέγραφος. Είναι ένα δυαδικό ψηφιακό σύστημα τηλεπικοινωνιών με κωδικοποίηση μεταβλητού μήκους λέξεων
1844	Λειτουργία πρώτης τηλεγραφικής γραμμής (Ουάσινγκτον - Βαλτιμόρη)
1876	A.G. Bell: Εφεύρεση τηλεφώνου
1887	Heinrich Hertz: Επαλήθευση της θεωρίας του Maxwell περί της ύπαρξης και διάδοσης ηλεκτρομαγνητικών κυμάτων
1893	Nikolai Tesla: Παρουσίασε και επέδειξε τις βασικές αρχές ενός ασύρματου τηλεπικοινωνιακού συστήματος
1895	Guglielmo Marconi: Εφηύρε και εφάρμοσε εμπορικά ένα σύστημα ασύρματης τηλεγραφίας
1906	Lee de Forest: Τρίοδος ενισχυτική λυχνία κενού. Κατέστησε δυνατή την εισαγωγή ενισχυτών σήματος στα τηλεφωνικά συστήματα και συνεπώς επέτρεψε την μετάδοση τους σε μεγαλύτερες αποστάσεις. Έπαιξε σημαντικό ρόλο στην ανάπτυξη της ραδιοφωνίας και των τηλεπικοινωνιών εν γένει
1920	Συντελέσθη η πρώτη ραδιοφωνική μετάδοση AM
1959	Kilby, Noyce: Ολοκληρωμένο κύκλωμα. Συνέβαλε στην ουσιαστική μείωση των φυσικών διαστάσεων, της κατανάλωσης ισχύος, του βάρους και του κόστους παραγωγής των τηλεπικοινωνιακών συστημάτων.

1962	Τέθηκε σε τροχιά από τη NASA ο πρώτος πειραματικός δορυφόρος Telstart I που μετέδιδε τηλεοπτικό σήμα
1979-1983	Πρώτη εμπορικά διαθέσιμη τεχνολογία αναλογικής (κινητής) κυψελωτής τηλεφωνίας (1G) στο Τόκιο, Φινλανδία, Νορβηγία, Σουηδία, Δανία και Σικάγο
1991	Ψηφιακή κυψελωτή κινητή τηλεφωνία δεύτερης γενιάς (2G) στη Φινλανδία και αλλού. Μέχρι σήμερα είναι η επικρατέστερη τεχνολογία κινητής τηλεφωνίας.
2001	Ψηφιακή κυψελωτή κινητή τηλεφωνία υψηλού ρυθμού μετάδοσης δεδομένων και πολυμέσων (3G) στο Τόκιο

Πίνακας 1.1: Σημαντικοί σταθμοί στην ιστορία των τηλεπικοινωνιών

Ψηφιακές τηλεπικοινωνίες



Σχήμα 1.1: Σχεδιάγραμμα ψηφιακού τηλεπικοινωνιακού συστήματος

Ένα σύστημα ψηφιακών επικοινωνιών, σαν και αυτό που πραγματεύεται το παρόν, απαρτίζεται από τα εξής μέρη (Σχήμα 1.1).

Από τη μεριά του **πομπού**:

- (1) πηγή της ψηφιακής πληροφορίας και μετατροπέας εισόδου
- (2) κωδικοποιητής πηγής
- (3) κωδικοποιητής καναλιού
- (4) ψηφιακός διαμορφωτής

Από τη μεριά του **δέκτη**:

- (5) ψηφιακός αποδιαμορφωτής
- (6) αποκωδικοποιητής καναλιού
- (7) αποκωδικοποιητής πηγής
- (8) μετατροπέας εξόδου

Φυσικά το υλικό ή άυλο μέσο που παρεμβάλλεται μεταξύ πομπού και δέκτη, το **κανάλι**, εκτός από υπόστρωμα μετάδοσης του μηνύματος είναι και πηγή θορύβου και παραμορφώσεων που εισάγονται στο μήνυμα. Σκοπός των συστημάτων επικοινωνίας είναι να μεταδώσουν ένα μήνυμα μέσα από ένα κανάλι, αναπαράγοντάς το όσο το δυνατόν πιο πιστά στην πλευρά του δέκτη.

1.1.1 Πομπός

Η **πηγή πληροφορίας** μπορεί να είναι ένα αναλογικό σήμα (φωνή, εικόνα, ένδειξη ενός αισθητήρα) ή ψηφιακή (δυναμικά αρχεία, κείμενο αποθηκευμένο σε υπολογιστή, SMS). Τα αναλογικά σήματα δειγματοληπτούνται από τον μετατροπέα εισόδου (συνήθως ένας Analog-to-Digital-Converter, ADC) και μετατρέπονται σε ψηφιακά. Ωστόσο επιδιώκουμε το ψηφιακό σήμα που θα μεταδοθεί να είναι χωρίς πλεονάζουσα πληροφορία. Με τη χρήση διαφόρων αλγορίθμων και θεωριών που προέρχονται κυρίως από την επιστήμη της πληροφορικής αλλά και από άλλους όπως η ψυχοακουστική, επιτυγχάνεται η συμπίεση των δεδομένων ή η **κωδικοποίηση πηγής**. Αυτό το στάδιο μπορεί να εισάγει απώλεια στην πληροφορία εισόδου, στο βαθμό που το σήμα δεν αλλοιώνεται αισθητά σε σχέση με το σκοπό για τον οποίο προορίζεται.

Η ψηφιακή ακολουθία που προέρχεται από αυτό το στάδιο εισάγεται στον **κωδικοποιητή καναλιού**. Εδώ προστίθεται κάποια πλεονάζουσα πληροφορία με σκοπό την ανίχνευση ή/και διόρθωση λαθών στην αποδιαμόρφωση του σήματος στο δέκτη. Με αυτόν τον τρόπο αυξάνεται η αξιοπιστία του συστήματος και επιτυγχάνεται εξαιρετική ανοχή στο θόρυβο και της αλλοιώσεις του σήματος.

Το επόμενο στάδιο της αλυσίδας του πομπού είναι ο **ψηφιακός διαμορφωτής**. Είναι η διεπαφή του συστήματος με το κανάλι. Οι ακολουθίες δυαδικής πληροφορίας μετατρέπονται, ή καλύτερα απεικονίζονται (mapping), σε διακριτές κυματομορφές ικανές να διασχίσουν το κανάλι με τη λιγότερη δυνατή παραμόρφωση. Το σχήμα και η διάρκειά τους είναι δύο από τους παράγοντες που παίζουν καθοριστικό ρόλο στην απόδοση και επίδοση του συστήματος.

Παρατήρηση Αξίζει να αναφερθεί πως στα ψηφιακά συστήματα τηλεπικοινωνιών, όταν έρθει η στιγμή της εκπομπής του διαμορφωμένου ψηφιακού σήματος, το σήμα μετατρέπεται και πάλι σε αναλογικό για να μπορέσει να διασχίσει το κανάλι του πραγματικού κόσμου, το οποίο δεν μπορεί παρά να είναι συνεχές, τόσο στο χρόνο όσο και στο πλάτος. Κατ' επέκταση οι κυματομορφές εξόδου του διαμορφωτή είναι (ή μετατρέπονται με τους Digital-to-Analog-Converters, DAC, σε) αναλογικά σήματα.

1.1.2 Τύποι διαμορφώσεων

Η εκάστοτε διαμόρφωση χαρακτηρίζεται από το αλφάβητο των **συμβόλων** που χρησιμοποιεί. Δηλαδή ένα σύνολο κυματομορφών, που η κάθε μία αντιστοιχεί σε μία ακολουθία δυαδικών ψηφίων (bit sequence). Ο πιο απλός τύπος διαμόρφωσης είναι η δυαδική διαμόρφωση, που χρησιμοποιεί δύο κυματομορφές και είναι ικανή να μεταδώσει ένα bit με κάθε σύμβολο. Στη γενική περίπτωση κατά την οποία έχουμε $M = 2^k$ κυματομορφές, με κάθε σύμβολο μεταδίδονται k bits. Αυτός ο τύπος καλείται M -αδική διαμόρφωση. Μία καθοριστική παράμετρος για το σύστημα είναι η χρονική διάρκεια ενός συμβόλου T_s ή η **περίοδος σηματοδοσίας**. Το αντίστροφο αυτής μέγεθος, είναι ο ρυθμός συμβόλων (baud rate) $R_s = 1/T_s$. Ένα παραγόμενο από τα παραπάνω μέγεθος που χαρακτηρίζει την απόδοση του συστήματος είναι ο ρυθμός μετάδοσης των δυαδικών ψηφίων της πληροφορίας (bit rate) $R_b = R_s k$.

Μια ακόμα διαφοροποίηση των διαμορφώσεων είναι : αυτές που το φάσμα του παραγόμενου σήματος περιλαμβάνει τη μηδενική συχνότητα (**διαμόρφωση βασικής ζώνης**) και αυτές που δεν την περιλαμβάνει (**ζωνοπερατή διαμόρφωση**). Δεδομένων M κυματομορφών βασικής ζώνης υπάρχουν διαφορετικοί τρόποι για την μετατροπή τους σε ζωνοπερατές κυματομορφές. Η βασική αρχή μετάδοσης τους είναι η διαμόρφωση ημιτονικών φερόντων αποτυπώνοντας την πληροφορία ενός συμβόλου είτε στη φάση, είτε στο πλάτος, είτε στη συχνότητα του φέροντος. Κάποιοι από αυτούς είναι η διαμόρφωση παλμών κατά

πλάτος (PAM), η διαμόρφωση παλμών κατά φάση (PSK), η ορθογώνια διαμόρφωση παλμών κατά πλάτος (QAM), η μεταλλαγή ολίσθησης συχνότητας (FSK).

Εκτός από τις ζωνοπερατές διαμορφώσεις ενός φέροντος υπάρχουν και αυτές **πολλαπλών φερόντων** (multicarrier modulation). Σε αυτές διαμορφώνονται πολλαπλά, παράλληλα φέροντα από την ίδια πηγή πληροφορίας και αντίστοιχα αποδιαμορφώνονται παράλληλα στο δέκτη. Παραδείγματα είναι η FDM και OFDM τεχνική διαμόρφωσης. Ο τελευταίος τύπος θα αναλυθεί πιο διεξοδικά στο Κεφ. 2 μαζί με τις βασικότερες αρχές του.

1.1.3 Δέκτης

Το λαμβανόμενο σήμα (αναλογικό) διέρχεται μέσω του **ψηφιακού αποδιαμορφωτή** και μετατρέπεται και πάλι σε ψηφιακό, δηλαδή σε ακολουθία δυαδικών ψηφίων διακριτού χρόνου. Με τη χρήση διαφόρων τεχνικών, γίνεται προσπάθεια ανάκτησης της μεταδιδόμενης ακολουθίας ψηφίων από το παραμορφωμένο ληφθέν σήμα.

Βασικές λειτουργίες του αποδιαμορφωτή είναι :

- η εκτίμηση του καναλιού (channel estimation),
- ο συγχρονισμός του δέκτη με τον πομπό
- η αντιστάθμιση σφαλμάτων στη φάση και τη συχνότητα του φέροντος
- η φώραση, δηλαδή η ανίχνευση του συμβόλου που έχει μεταδοθεί (από το σύνολο των M δυνατών συμβόλων) σε κάθε περίοδο σηματοδοσίας και κατ' επέκταση η απόφαση της ακολουθίας k bits που μεταδόθηκε.

Από την παραπάνω διαδικασία, ανάλογα με την κωδικοποίηση καναλιού που έχει εφαρμοστεί στα προς μετάδοση δεδομένα, μπορεί να μην προκύψει κάποια απόφαση ή να προκύψει λανθασμένη.

Σε αυτό το σημείο ο **κωδικοποιητής καναλιού** γνωρίζοντας τον κώδικα που έχει χρησιμοποιηθεί και τις παραμέτρους του, εφαρμόζει διάφορους αλγορίθμους πάνω στα μπλοκ των k bits ανιχνεύοντας τα εσφαλμένα bit και προσπαθώντας να τα ανακτήσει. Αυτό είναι δυνατόν, λόγω της πλεονάζουσας πληροφορίας που είχε εισαχθεί κατά τη μετάδοση από τον κωδικοποιητή καναλιού. Φυσικά, δεν είναι δυνατό να ανακτηθεί πλήρως η πληροφορία για κάθε συνθήκη του καναλιού. Σε αυτές τις περιπτώσεις, τα δεδομένα στην έξοδο του αποκωδικοποιητή διαφέρουν από τα αντίστοιχα στην είσοδο του κωδικοποιητή. Η **μέση πιθανότητα σφάλματος** (Bit error rate, BER) είναι ένα μέτρο επίδοσης του συνδυασμού αποδιαμορφωτή-αποκωδικοποιητή. Παράγοντες που επηρεάζουν το BER είναι :

- τα χαρακτηριστικά του καναλιού
 - η ισχύς θορύβου
 - το είδος παρεμβολών
 - η συνάρτηση μεταφοράς)
- η κωδικοποίηση καναλιού
- ο τύπος διαμόρφωσης και αποδιαμόρφωσης
- η μορφή των κυματομορφών (συμβόλων)
- η ισχύς του πομπού

Το τελικό στάδιο της αλυσίδας είναι η **αποκωδικοποίηση πηγής** ή η αποσυμπίεση των δεδομένων. Στην έξοδο αυτού του σταδίου ανακτάται η αρχική ψηφιακή πληροφορία της πηγής (πρωτού αυτή υποστεί συμπίεση). Εάν το αρχικό μήνυμα είναι σε ψηφιακή μορφή (π.χ. κείμενο), αυτή είναι η έξοδος του δέκτη. Εάν η αρχική μορφή είναι αναλογική, ακολουθεί και το στάδιο του μετατροπέα εξόδου (συνήθως ένας DAC), που ανακατασκευάζει το αναλογικό σήμα (π.χ. ήχος) από το ψηφιακό. Η διαφορά του ληφθέντος σήματος από αυτό που μεταδόθηκε είναι η παραμόρφωση που εισάγεται από το σύστημα ψηφιακής μετάδοσης.

1.1.4 Κανάλι

Ως τηλεπικοινωνιακό κανάλι νοείται το φυσικό μέσο το οποίο παρεμβάλλεται ανάμεσα στον πομπό και το δέκτη. Για τις ασύρματες επικοινωνίες (μικροκύματα, μεσαία, βραχέα κύματα) το κανάλι είναι ο ατμοσφαιρικός αέρας, ο κενός χώρος στο διάστημα, το έδαφος ή το νερό της θάλασσας. Για τις ενσύρματες, το κανάλι μπορεί να είναι ένα ζεύγος συρμάτων χαλκού ή μία οπτική ίνα. Σε όλες τις περιπτώσεις το σήμα παραμορφώνεται κατά τυχαίο τρόπο με :

- α) προσθετικές παραμορφώσεις, όπου στο μεταδιδόμενο σήμα προστίθεται κάποια τυχαία διαδικασία θορύβου
- β) μη προσθετικές παραμορφώσεις όπου το κανάλι δρα συνελικτικά πάνω στο μεταδιδόμενο σήμα προκαλώντας πιο περίπλοκες αλλοιώσεις.

Ο **προσθετικός θόρυβος** μπορεί να είναι θερμικός θόρυβος προερχόμενος από την άτακτη κίνηση των ηλεκτρονίων κυρίως στην πρώτη βαθμίδα του δέκτη. Μπορεί επίσης να είναι θόρυβος παρεμβολών από άλλους χρήστες του καναλιού. Για τις ασύρματες μεταδόσεις προσθετικός θόρυβος μπορεί να προκαλείται επιπλέον από ανθρώπινες δραστηριότητες (π.χ. το άναμμα μιας μηχανής) ή από φαινόμενα της ατμόσφαιρας (π.χ. ηλεκτρικές εκκενώσεις κεραυνών).

Ο **μη προσθετικός θόρυβος** εμφανίζεται κυρίως σε ασύρματα κανάλια. Μία από τις πιο συχνά εμφανιζόμενες μορφές του, τόσο στις ζεύξεις μεγάλων αποστάσεων με βραχέα κύματα μέσω της ιονόσφαιρας, όσο και σε μικροκυματικές ζεύξεις σε αστικό περιβάλλον, είναι η **πολυδιαδρομική διάδοση** (multipath propagation). Το αποτέλεσμα της είναι χρονικές διακυμάνσεις στην ένταση του λαμβανόμενου σήματος, οι λεγόμενες διαλείψεις (fading). Ισοδύναμα στο πεδίο των συχνοτήτων, τέτοια κανάλια παρουσιάζουν συχνοεπιλεκτική συμπεριφορά (frequency selectivity).

Οι προαναφερθέντες τύποι παραμορφώσεων παρουσιάζουν στατιστική συμπεριφορά, που μπορεί να μοντελοποιηθεί με μαθηματικό τρόπο και να ληφθεί υπόψη στο σχεδιασμό του συστήματος. Ωστόσο, τα στατιστικά χαρακτηριστικά μπορεί να είναι στατικά ή χρονικά μεταβαλλόμενα. Στην τελευταία περίπτωση, απαιτείται η ύπαρξη προσαρμοστικότητας στα αντίστοιχα συστήματα, για να ανταποκρίνονται στις αλλαγές του καναλιού.

Ένα βασικό χαρακτηριστικό του καναλιού είναι το **εύρος ζώνης** (φάσμα) διέλευσης καθώς και η κεντρική συχνότητα του φάσματος αυτού. Δηλαδή το κομμάτι εκείνο του φάσματος των ηλεκτρομαγνητικών κυμάτων που είναι διαθέσιμο για την μετάδοση του σήματος χωρίς ιδιαίτερη εξασθένιση. Διακρίνουμε λοιπόν τα κανάλια, όπως και τα αντίστοιχα σήματα και τις διαμορφώσεις, σε κανάλια βασικής ζώνης (περιλαμβάνουν τη μηδενική συχνότητα) και ζωνοπερατά κανάλια (βρίσκονται φασματικά γύρω από μία κεντρική μηδενική συχνότητα).

Συγκριτικά με τα αναλογικά

Η βασική διαφορά μεταξύ ενός συστήματος ψηφιακής μετάδοσης από ένα αναλογικό είναι πως η αρχική πληροφορία, ή το αρχικό μήνυμα, είναι εξ αρχής (ή μετατρέπεται μέσω μίας διάταξης ADC σε) ψηφιακό σήμα, δηλαδή σε μία ακολουθία (διακριτού χρόνου) δυαδικών ψηφίων που αναπαριστούν το αρχικό μήνυμα.

Ένα κύριο πλεονέκτημα των ψηφιακών συστημάτων (τηλεπικοινωνιακών ή μη) έναντι των αναλογικών ισοδυναμίων τους, είναι η πλήρης ανοχή στο θόρυβο. Το μήνυμα μπορεί να αναγεννηθεί πλήρως, στην αρχική ψηφιακή του αναπαράσταση στον πομπό, μετά από κάθε στάδιο αναμετάδοσης, αφαιρώντας την επίδραση του θορύβου. Στα αναλογικά συστήματα, ο θόρυβος προστίθεται και είναι παρών μετά από κάθε προσπάθεια ενίσχυσης ή ανάκτησης του αρχικού σήματος και είναι πολύ δύσκολο να αφαιρεθεί. Επιπλέον, οι συνήθεις πηγές (αναλογικές ή ψηφιακές) φέρουν αρκετή πλεονάζουσα πληροφορία, η οποία μέσω τεχνικών ψηφιακής επεξεργασίας σήματος (DSP) δύναται να μειωθεί ή να αφαιρεθεί. Συνεπώς, γίνεται αποτελεσματικότερη χρήση του διαθέσιμου εύρους ζώνης του καναλιού. Ακόμα, με την συνεχή εξέλιξη της τεχνολογίας πάνω στα ψηφιακά συστήματα και τη μαζική παραγωγή υπολογιστών ειδικού σκοπού (DSP processors) και ψηφιακών κυκλωμάτων ειδικού σκοπού (ASIC), η κατασκευή των ψηφιακών συστημάτων επικοινωνίας είναι πιο οικονομική έναντι των αναλογικών. Τέλος, για σύνθετες τεχνικές ψηφιακής διαμόρφωσης όπως το OFDM, η πρόοδος της ψηφιακής τεχνολογίας (π.χ. VLSI) έχει, πρακτικά, ανοίξει το δρόμο για την υλοποίησή τους.

1.2 Περί Ψηφιακών Συστημάτων

1.2.1 Απο τη Θεωρία στην Πράξη

Δεδομένης της εξεζητημένης πολυπλοκότητας των σύγχρονων ψηφιακών συστημάτων, όπως το σύστημα ψηφιακής επικοινωνίας που αναφέρθηκε στο προηγούμενο τμήμα, εύλογα τίθεται το ερώτημα με ποιό τρόπο καθίσταται δυνατή η υλοποίησή τους. Η προφάνεια της απάντησης (τεχνολογία ψηφιακών ηλεκτρονικών), περιορίζεται αν αναλογιστεί κανείς τη μη γραμμική εξελικτική πορεία της εν λόγω τεχνολογίας.

Η επιτυχής εφαρμογή των αρχικών ανακαλύψεων στον τομέα της κβαντικής φυσικής και η συστηματοποίηση των γνώσεων που αφορούν στα ηλεκτρικά κυκλώματα οδήγησαν στην εποχή των αναλογικών ηλεκτρονικών. Εφευρέσεις όπως η δίοδος, η λυχνία κενού, το τρανζίστορ, ο τελεστικός ενισχυτής έδωσαν μεγάλη ώθηση στην ανάπτυξη της τεχνολογίας εν γένει. Παρόλα αυτά υπήρχε ακόμη ένα άνω όριο ως προς την υλοποίηση πολύπλοκων συστημάτων, αφ' ενός λόγω περιορισμένης ακρίβειας και χαμηλής ταχύτητας στην αναλογική επεξεργασία σημάτων και αφετέρου, λόγω των ολοένα και μεγαλύτερων φυσικών διαστάσεων των κυκλωμάτων. Αυτό το άνω όριο το κατέρριψαν δυο ιστορικές ανακαλύψεις που, δικαιολογημένα, έχουν θέση στο πάνθεον των κατακτήσεων του ανθρώπινου πνεύματος :

- Τα θεωρήματα δειγματοληψίας και εύρους ζώνης του Shannon (1948), ο οποίος επεξέτεινε την εργασία του Nyquist, τα οποία γέννησαν την ψηφιακή επεξεργασία σήματος οριοθετώντας, έτσι, την απαρχή της ψηφιακής εποχής
- Η δημιουργία του πρώτου ολοκληρωμένου κυκλώματος το 1954.

Ο πίνακας 1.2 παρουσιάζει, συνοπτικά, γεγονότα-σταθμούς στην εξέλιξη της (ψηφιακής) ηλεκτρονικής τεχνολογίας και καταδुकνεύει το μη-αναπόφευκτο εντέλει της σύγχρονης έκρηξης της τεχνολογίας της πληροφορίας.

1904	<i>Sir Fleming</i>	Δίοδος (Diode, Thermionic Valve)
1907	<i>De Forest</i>	Τρίοδος (Audion)
1940	<i>Ohl</i>	Ohl's Crystal - πρόδρομος του φωτοβολταϊκού στοιχείου και προπομπός του transistor - Βασίζεται στην παρατήρηση ότι : <i>"Οι ακαθαρσίες (impurities) σε ημιαγώγιμους κρυστάλλους εμφανίζουν φωτοηλεκτρικές ιδιότητες"</i>
1947	<i>Bardeen, Brattain, Shockley [Bell Labs]</i>	Το πρώτο transistor

1954		Μαζική παραγωγή του πρώτου βασισμένου-σε-πυρίτιο transistor
1954	<i>Texas Instruments</i>	Το πρώτο φορητό "τρανζιστοράκι". Αντικατάσταση της τριόδου λυχνίας από το transistor
1955	<i>Frosch, Derick [Bell Labs]</i>	Χρησιμοποίηση του SiO_2 για την παραγωγή επαφών <i>PN</i> με την τεχνική της διάχυσης . Αντικατάσταση του γερμανίου από το πυρίτιο ως προτιμώμενος ημιαγωγός για τη βιομηχανία ηλεκτρονικών
1958	<i>Kilby [TI], Noyce [Fairchild]</i>	Εφεύρεση του ολοκληρωμένου κυκλώματος (<i>Integrated Circuit, IC</i>)
1961	<i>Noyce [Fairchild]</i>	Resistor Transistor Logic (RTL). Με βάση αυτή την τεχνολογία παρουσιάστηκε το πρώτο εμπορικό ολοκληρωμένο κύκλωμα. Επρόκειτο για ένα <i>SR Flip-Flop</i> .
1961	<i>Thomson [Plessey Labs]</i>	Transistor Transistor Logic (TTL). Τεχνολογία που επικράτησε στη βιομηχανία μικροηλεκτρονικής μέχρι τη δεκαετία του 70 οπότε και αντικαταστάθηκε από τα <i>CMOS</i> .
1962	<i>Hofstein, Heiman [RCA]</i>	Εφεύρεση του MOSFET . Αν και πιο αργό από το <i>BJT</i> , είναι μικρότερο, φθηνότερο και καταναλώνει λιγότερη ισχύ, γεγονός που επιτρέπει τη συσσώματωση ακόμα μεγαλύτερου αριθμού transistor χωρίς να προκύψει πρόβλημα υπερθέρμανσης του <i>chip</i> . Βασικό δομικό στοιχείο των περισσότερων μικροεπεξεργαστών.
1963	<i>Wanlass, Sah [Fairchild]</i>	Complementary Metal-Oxide-Semiconductor (CMOS). Τα <i>CMOS</i> transistor ελάττωσαν κατακόρυφα τις ενεργειακές απαιτήσεις και, σε συνδυασμό με τις μικρές διαστάσεις τους, άνοιξαν το δρόμο για την <i>Ολοκλήρωση-Πολύ-Μεγάλης-Κλίμακος (VLSI)</i> . Αποτελούν τη βάση της συντριπτικής πλειοψηφίας των ολοκληρωμένων κυκλωμάτων.
1965	<i>Lucky [Bell Labs]</i>	Εφεύρεση του <i>Αυτόματου Προσαρμοστικού Ισοσταθμιστή (Automatic Adaptive Equalizer)</i> . Οι ΑΑΕ διορθώνουν αλλοιωμένα σήματα βελτιώνοντας σημαντικά την ταχύτητα και την απόδοση. Χρησιμοποιούνται ακόμη και σήμερα στα <i>modem</i> .
1967	<i>Texas Instruments</i>	Ο πρώτος υπολογιστής χειρός ("κομπιουτεράκι")
1969	<i>Dennard</i>	Dynamic RAM (DRAM) - 1 transistor DRAM. Έδωσε μεγάλη ώθηση στην τεχνολογία μνημών επιτρέποντας πυκνότερες μνήμες (και αρα μεγαλύτερες χωρητικότητες) και αυξημένη ταχύτητα. Βασικό συστατικό όλων των υπολογιστών.
1971	<i>Faggin, Hoff, Mazor [Intel]</i>	Computer-on-Chip (COC) . Ο πρώτος υπολογιστής σε ένα chip, ο 4-μπιτος 4004, εκτελεί 60000 λειτουργίες το δευτερόλεπτο.
1974	<i>Texas Instruments</i>	TMS 1000 . Ο πιο πολυχρησιμοποιημένος <i>COC</i> με εφαρμογές όπως: παιχνίδια, κομπιουτεράκια, φωτοτυπικά μηχανήματα, συσκευές, συναγερμοί κλπ.
1977	<i>Birkner, Chua [Monolithic Memories]</i>	Programmable Logic Device (PLD)*
1978	<i>Bredlove, Wiggins, Frantz [TI]</i>	Digital Signal Processor (DSP, ΨΕΣ)*
1980	<i>Mead, Conway</i>	VLSI (Ολοκλήρωση-Πολύ-Μεγάλης-Κλίμακος, Very-Large-Scale-Integration Circuits)*
1984	<i>Freeman [Xilinx]</i>	Field Programmable Gate Array (FPGA)*

Πίνακας 1.2: Συνοπτική αναδρομή σε σημεία σταθμούς στην ιστορία της ηλεκτρονικής

1.2.2 Ψηφιακά Συστήματα - Σχεδίαση

Τα λογικά κυκλώματα αποτελούν το στοιχείο-κλειδί (*key element*) των ηλεκτρονικών υπολογιστών και παράλληλα χρησιμοποιούνται σε μια ευρύτατη ποικιλία εφαρμογών. Όπως έχει ήδη αναφερθεί, τα λογικά κυκλώματα υλοποιούνται ηλεκτρονικά με τη χρήση transistor που υπάρχουν στα ολοκληρωμένα κυκλώματα (*ICs, chips*). Με τη σύγχρονη τεχνολογία VLSI¹ είναι εφικτή η δημιουργία chip που περιέχουν εκατομμύρια transistor. Οι εκτιμήσεις για το τρέχον έτος (2011) αναφέρουν αριθμό transistor που προσεγγίζει το 1 δισεκατομμύριο.²

Τώρα, πριν την ανάπτυξη της (γενικευμένης) μεθόδου σχεδίασης ΨΣ, ακολουθεί μια σύντομη περιγραφή των διαφόρων ειδών IC που μπορούν να χρησιμοποιηθούν ως τεχνολογία υλοποίησης.

Τυπικά Ολοκληρωμένα Κυκλώματα (Standard IC)

Κάθε IC αυτής της κατηγορίας είναι στην παραγωγικότητα ένα μικρο κύκλωμα (<100 transistor) και εκτελεί μια απλή εργασία. Η λειτουργία και η φυσική τους οργάνωση είναι προτυποποιημένη. Για την κατασκευή ενός σύνθετου κυκλωματος ο σχεδιαστής επιλέγει τα δομικά στοιχεία (standard IC's) και κατόπιν καθορίζει τη συστημική διασύνδεσή τους. Ένα βασικό μειονέκτημα της σύνθεσης ΨΣ με χρήση standard IC's είναι η σπατάλη πολύτιμου χώρου στην τελική πλακέτα (Printed Circuit Board - PCB) λόγω τοποθέτησης IC χαμηλής πολυπλοκότητας (και άρα περιορισμένων δυνατοτήτων). Άλλο μειονέκτημα είναι η περιορισμένη ευελιξία στη σχεδίαση καθώς το κάθε Standard IC έχει συγκεκριμένη λειτουργία που δε μπορεί να τροποποιηθεί.

Διατάξεις Προγραμματιζόμενης Λογικής (Programmable Logic Devices - PLD)

Σε αντίθεση με τα συγκεκριμένης λειτουργίας standard IC's, τα PLD περιέχουν κυκλώματα τα οποία μπορεί να οργανώσει ο χρήστης κατά το δοκούν. Βασικό τους πλεονέκτημα είναι η πολύ γενική δομή τους, που βασίζεται σε ένα σύνολο προγραμματιζόμενων λογικών διακοπών. Αυτή η δομή προσφέρει μεγάλη ευελιξία κατά τη σχεδίαση : οποιαδήποτε λειτουργία μπορεί να υλοποιηθεί με την κατάλληλη οργάνωση των διακοπών, μέσω του προγραμματισμού τους από τον τελικό χρήστη (end user). Οι περισσότεροι τύποι PLD είναι επαναπρογραμματιζόμενοι και ως εκ τούτου απλοποιούν κατά πολύ τη διαδικασία σχεδίασης και ελέγχου. Τυπικοί εκπρόσωποι αυτής της οικογένειας IC είναι οι *Διατάξεις Προγραμματιζόμενης Λογικής* (Programmable Logic Arrays - PLA) και οι *Διατάξεις Πυλών Επαναπρογραμματιζόμενου Πεδίου* (Field Programmable Gate Arrays - FPGA). Θα ακολουθήσει, σε επόμενη ενότητα, λεπτομερής περιγραφή της δομής και του τρόπου λειτουργίας των FPGA. Ενδεικτικά, πάντως, αναφέρεται ότι η ολοκλήρωση των FPGA

¹Το 1980, ο Αμερικανός φυσικός *Carver Mead* του Ινστιτούτου CalTech και η επιστήμων υπολογιστών *Lynn Conway*, άρτι απολυθείσα από μια υπερσυντηρητική IBM, συνέγραψαν το εγχειρίδιο *Introduction to VLSI Systems* το οποίο σύντομα αποτέλεσε την κύρια αναφορά των σχεδιαστών συστημάτων VLSI. Ο Mead, το 1972, πρόέβλεψε ότι η σμίκρυνση των transistor θα έφτανε την τάξη των 0.15μm : μια τάξη μεγέθους υπερβολικά μικρότερη της τεχνολογίας αιχμής εκείνης της περιόδου, των 10μm. Αφιέρωσε τα επόμενα χρόνια στην ανάπτυξη της απαραίτητης τεχνολογίας για την επίτευξη αυτού του "μικροσκοπικού" στόχου. Καταλυτικό στοιχείο στην επιτυχία της προσπάθειάς του, αποτέλεσε η ανάπτυξη του μεταγλωττιστή πυριτίου (silicon compiler), μια εφαρμογή CAD (Computer Aided Design), ανάλογη με το μεταγλωττιστή λογισμικού (software compiler). Αυτός επέτρεπε στο σχεδιαστή να καθορίσει της λειτουργίες του ολοκληρωμένου κυκλώματος (chip) με μία εύχρηστη και ευκολονόητη δομημένη γλώσσα προγραμματισμού. Ο παραγόμενος κώδικας, στη συνέχεια, μετατρέποταν από τον υπολογιστή στα τμήματα (tracks) που συναποτελούσαν το κάθε επίπεδο (layer) του κυκλωματος πυριτίου. Αυτά, με τη σειρά τους, τροφοδοτούσαν σε ένα σύστημα εκτύπωσης υψηλής ανάλυσης (plotter) το οποίο παρείχε τις δομές χάραξης για τη φυσική κατασκευή του chip. Αυτή η τεχνολογία, εκτός από το να παράσχει τα απαραίτητα εργαλεία για μια νέα γενιά μικροεπεξεργαστών και σύνθετων διατάξεων, ενθάρρυνε, επίσης, τη δημιουργία καινοτόμων εταιριών ημιαγωγών, *σχεδίασης-μόνο* (fabless), οι οποίες υποστηρίζονταν από εξειδικευμένα, για μέταλλα ημιαγωγών, χυτήρια. Κάποιες από τις εταιρείες αυτού του τύπου, διαδραματίζουν σήμερα, σπουδαίο ρόλο στην ανάπτυξη της βιομηχανίας ψηφιακών ηλεκτρονικών (Qualcomm, nVidia, ATI Technologies, Broadcom, AMD)

²<http://www.foxnews.com/scitech/2011/01/04/years-later-does-moores-law-hold-true/>

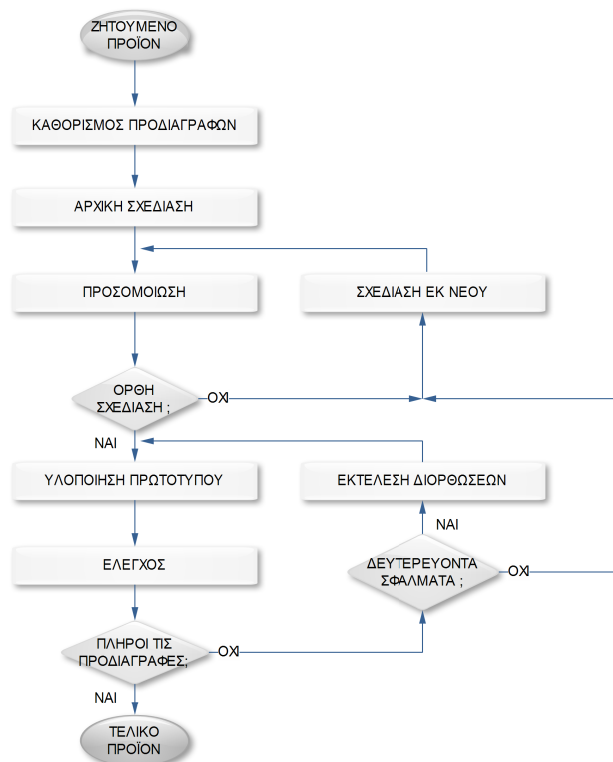
είναι της τάξης των δεκάδων εκατομμυρίων transistor.

Εξειδικευμένα Ολοκληρωμένα (Application Specific Interated Circuits - ASIC)

Είναι κυκλώματα που προορίζονται για χρήση σε ειδικές εφαρμογές. Έχουν το πλεονέκτημα, σε σχέση με τα PLD, ότι αποφεύγουν τη σπατάλη πολύτιμου χώρου λόγω των προγραμματιζόμενων διακοπών οι οποίοι, επιπλέον, περιορίζουν την ταχύτητα λειτουργίας του τελικού κυκλώματος. Η σχεδίασή τους δύναται να βελτιστοποιηθεί προς όφελος κάποιας λειτουργίας/χαρακτηριστικού και αυτό συνεπάγεται, συνήθως, καλύτερη απόδοση. Αν και η σχεδίαση του λογικού κυκλώματος είναι αποκλειστική ευθύνη του μηχανικού-σχεδιαστή, το τελικό προϊόν (IC) υλοποιείται από εταιρίες που διαθέτουν τον κατάλληλο εξοπλισμό, γεγονός που αυξάνει το κόστος του IC, ενώ, αρκετές φορές, η διαδικασία κατασκευής του απλώνεται σε μια μεγάλη χρονική περίοδο (της τάξης μηνών). Εδώ υπερέχουν σαφώς τα PLD, καθώς ο προγραμματισμός του IC γίνεται από τον τελικό χρήστη, χωρίς την παρεμβολή κάποιας εταιρίας και της αναπόφευκτης χρονικής καθυστέρησης. Εντέλει, τα ASIC κυκλώματα χρησιμοποιούνται σε κρίσιμες (critical) εφαρμογές, όπου η βελτιστοποίηση στο χώρο ή στην ταχύτητα είναι καθοριστική για τη σωστή λειτουργία και απόδοση του κυκλώματος.

Τώρα, επιστρέφοντας στη διαδικασία σχεδίασης, ενώ τα βασικά δομικά στοιχεία των λογικών κυκλωμάτων είναι εύκολα κατανοητά, δε συμβαίνει το ίδιο σε κυκλώματα που περιέχουν δεκάδες εκατομμύρια transistor. Μόνος αρωγός είναι η χρήση τεχνικών σχεδίασης υψηλής οργάνωσης.

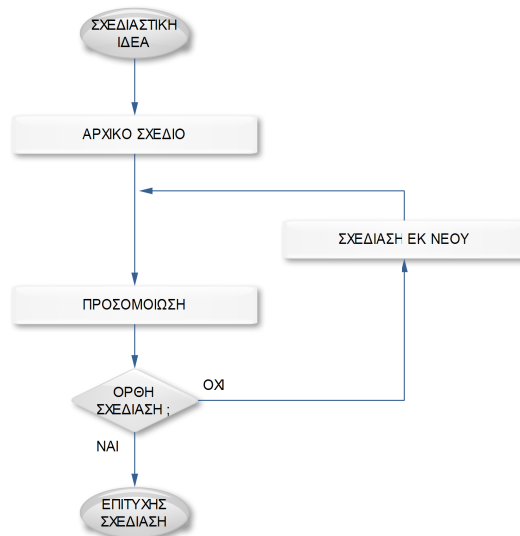
Η ευρεία διάδοση εργαλείων CAD (Computer-Aided-Design) έχει προτυποποιήσει, κατα μια έννοια, τη γενική διαδικασία σχεδίασης προϊόντων. Το διάγραμμα ροής του σχήματος 1.2 παρουσιάζει μια τυπική διαδικασία ανάπτυξης ενός προϊόντος. Το τελικό προϊόν θα μπορούσε να είναι κάλλιστα ένας φούρνος μικροκύματων, ένα περίπλοκο μηχανολογικό εξάρτημα, ένα ψηφιακό σύστημα κλπ.



Σχήμα 1.2: Γενική Διαδικασία Σχεδίασης Προϊόντων

Συγκεκριμενοποιώντας, τώρα, τη διαδικασία σχεδίασης για το ψηφιακό σύστημα, κρίνεται σκόπιμο να εισαχθεί η έννοια της *βαθμίδας* (unit) και του *συστήματος* (system). Κάθε βαθμίδα επιτελεί μια συγκεκριμένη λειτουργία και αποτελεί, κατά μια έννοια, ένα "γρανάζι" στον ωρολογιακό μηχανισμό του συστήματος. Το σύνολο των βαθμίδων καθώς και η *διεπαφή* που χαρακτηρίζει τη μεταξύ τους επικοινωνία (interface) συναποτελούν το σύστημα.

Ο βασικός βρόχος σχεδίασης (σχήμα 1.3) αναφέρεται στη βαθμίδα. Σημειώνεται ότι η βαθμίδα μπορεί να

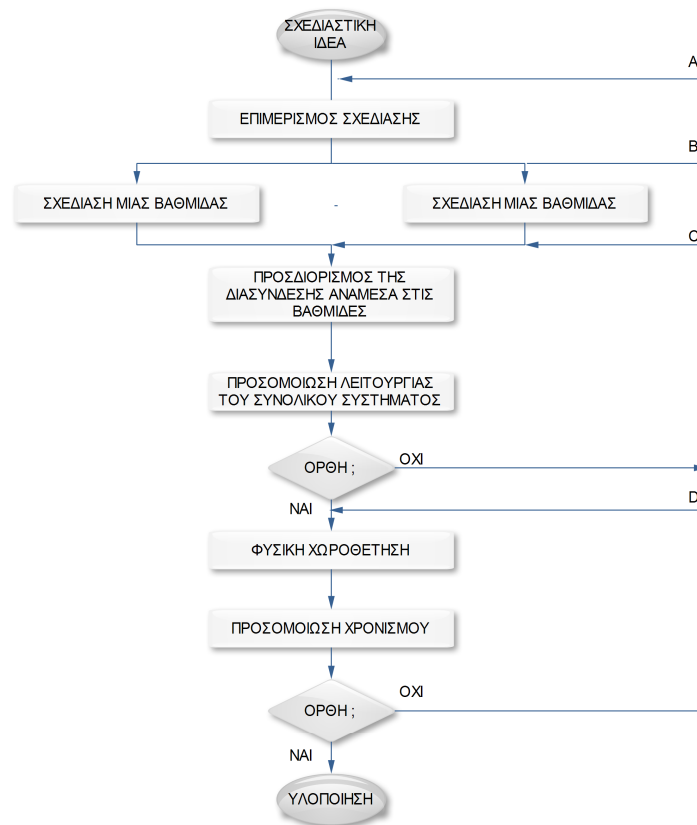


Σχήμα 1.3: Σχεδίαση Βαθμίδας

είναι ένα μικρό σύστημα από μόνη της (υπο-σύστημα), αποτελούμενη από διάφορες βαθμίδες χαμηλότερης πολυπλοκότητας και την εσωτερική τους διεπαφή. Το διάγραμμα ροής του σχήματος 1.4 παρουσιάζει τον πλήρη κύκλο ανάπτυξης ενός ψηφιακού συστήματος.

Επεξηγήσεις για τα διαγράμματα ροής

- Η προσομοίωση λειτουργίας (functional simulation) εξασφαλίζει ότι η υπο ανάπτυξη βαθμίδα/σύστημα εκτελεί όλες τις προβλεπόμενες λειτουργίες χωρίς παρενέργειες.
- Η φυσική χωροθέτηση (physical place and route) είναι ο προσδιορισμός του σημείου τοποθέτησης των δομικών συστατικών του συστήματος καθώς και η μορφή των, απαραίτητων για τη σωστή αναμεταξύ τους διασύνδεση, αγωγίμων συνδέσεων. Η εργασία αυτή εκτελείται, πλέον, αυτόματα με χρήση προγραμμάτων CAD.
- Η προσομοίωση χρονισμού (timing simulation) αποκαλύπτει ενδεχόμενα προβλήματα στην ταχύτητα λειτουργίας του κυκλώματος. Αυτά τα προβλήματα προκαλούνται λόγω της ύπαρξης πραγματικών (και όχι ιδανικών) αγωγίμων συνδέσεων στο φυσικό κύκλωμα, και εμφανίζουν αντίσταση



Σχήμα 1.4: Κύκλος Ανάπτυξης Ψηφιακού Συστήματος

και παρασιτικές χωρητικότητες κατά τη διέλευση του ηλεκτρικού σήματος. Τα προβλήματα χρονισμού διορθώνονται με τη βοήθεια των προγραμμάτων CAD μέσω φυσικής αναδιάταξης των δομικών στοιχείων της πλακέτας.

1.2.3 Δομικά Στοιχεία Συστημάτων CAD - Γλώσσες Περιγραφής Υλικού - Μέρος Α

Σε αυτό το τμήμα θα παρουσιαστούν με κάπως περισσότερη λεπτομέρεια τα συστήματα CAD που αφορούν στη σχεδίαση των ψηφιακών συστημάτων και θα γίνει μια πρώτη αναφορά και παρουσίαση των *Γλωσσών Περιγραφής Υλικού* (Hardware Description Languages - HDL), δύο εξέχοντα μέλη της οποίας είναι οι γλώσσες *VHDL* και *Verilog*.

Όπως αναφέρθηκε προηγουμένως προκειμένου να καταστεί δυνατή η σχεδίαση και υλοποίηση πολύπλοκων ψηφιακών συστημάτων απαιτείται ένας αριθμός εργαλείων CAD. Αυτά συνήθως βρίσκονται μαζί, σε ένα ενοποιημένο περιβάλλον που αποκαλείται σύστημα σχεδίασης CAD. Παραδείγματα είναι το σύστημα Quartus της Altera, το ISE της Xilinx κλπ.

Τα στάδια μιας τυπικής διαδικασίας σχεδίασης είναι :

1. Εισαγωγή Σχεδίασης
 - Πίνακας Αληθείας
 - Σχηματικό Διάγραμμα
 - Περιγραφή HDL
2. Αρχική Σύνθεση

3. Λειτουργική Προσομοίωση
4. Λογική Σύνθεση & Βελτιστοποίηση
5. Φυσική Χωροθέτηση
6. Προσομοίωση Χρονισμού
7. Οργάνωση IC

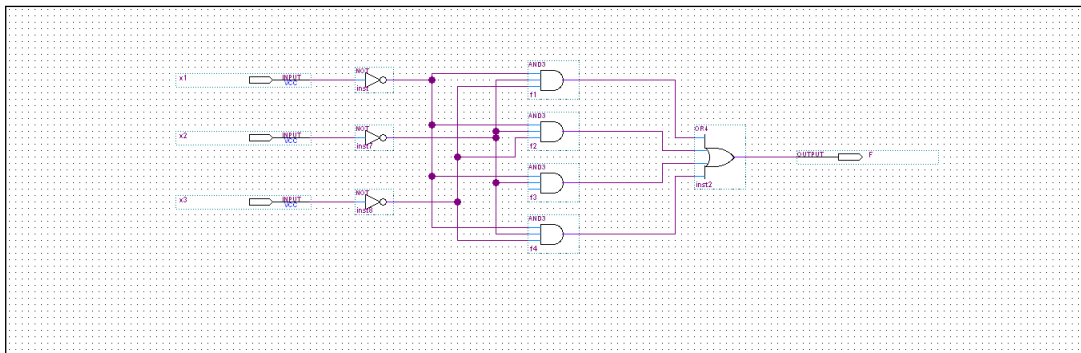
Εισαγωγή Σχεδίασης

Αυτό το βήμα εκτελείται από το σχεδιαστή και είναι η περιγραφή και μορφοποίηση της γενικής δομής του κυκλώματος. Προϋποθέτει την πλήρη κατανόηση της λειτουργίας που θα πρέπει να επιτελεί το κύκλωμα. Δεν αυτοματοποιείται γιατί στηρίζεται στις ανθρώπινες ιδιότητες της εμπειρίας και της διαίσθησης. Ακολουθούν τρεις μέθοδοι εισαγωγής σχεδίασης, με την πρώτη να είναι ιδανική για κυκλώματα μικρής πολυπλοκότητας, τη δεύτερη για κυκλώματα μεσαίας πολυπλοκότητας και την τρίτη για κυκλώματα/συστήματα υψηλής πολυπλοκότητας.

Πίνακας Αληθείας

Στηρίζεται στο γεγονός ότι κάθε λογική συνάρτηση μπορεί να περιγραφεί πλήρως με τη βοήθεια ενός πίνακα αληθείας. Η περιγραφή εισάγεται στο σύστημα με ένα απλό αρχείο κειμένου. Ισοδύναμη είναι η χρήση του *επεξεργαστή κυματομορφών* (waveform editor) του συστήματος για την παραγωγή του διαγράμματος χρονισμού (μορφή αναπαράστασης πληροφορίας πλήρως ισοδύναμη και εναλλάξιμη με τον πίνακα αληθείας). Το CAD αυτόματα μετασχηματίζει τις πιο πάνω περιγραφές σε κύκλωμα λογικών πυλών.

Σχηματικό Διάγραμμα



Σχήμα 1.5: Χρήση του Graphical Editor (Altera Quartus II CAD)

Διαγράμματα σαν αυτό παράγονται από το *Γραφικό Επεξεργαστή* (Graphical Editor) του συστήματος CAD. Τα στοιχεία του κυκλώματος απεικονίζονται ως γραφικά σύμβολα και οι ευθείες γραμμές αναπαριστούν τις μεταξύ τους συνδέσεις. Το σχηματικό διάγραμμα μετασχηματίζεται απευθείας σε κύκλωμα λογικών πυλών με τη χρήση του *Schematic Capture Tool* του συστήματος CAD.

Γλώσσες Περιγραφής Υλικού

Όταν οι απαιτήσεις πολυπλοκότητας του υπό ανάπτυξη ψηφιακού συστήματος είναι αυξημένες, οι δυο προηγούμενες μέθοδοι δεν αποτελούν ρεαλιστικές επιλογές. Σε αυτή την περίπτωση η περιγραφή του

κυκλώματος δίνεται με τη βοήθεια μιας Γλώσσας Περιγραφής Υλικού (Hardware Description Language - HDL).

Μια HDL είναι μια κατ'ανάγκη γλώσσα υψηλού επιπέδου, που φέρει αρκετές ομοιότητες με μια τυπική γλώσσα προγραμματισμού υπολογιστών. Η διαφορά έγκειται στο ότι χρησιμοποιείται για να περιγράψει ένα κύκλωμα και όχι ένα πρόγραμμα το οποίο είναι προορισμένο να εκτελεστεί από κάποιον υπολογιστή. Ως εκ τούτου χρησιμοποιείται ο όρος κώδικας μάλλον, παρά πρόγραμμα (π.χ. κώδικας VHDL και όχι πρόγραμμα VHDL). Παραδείγματα γλωσσών είναι η VHDL και η Verilog, και οι δυο προτυποποιημένες και ευρύτατα χρησιμοποιούμενες στη βιομηχανία. Μιλώντας όσο πιο γενικά γίνεται (λεπτομερής αναφορά στη γλώσσα VHDL - η οποία επιλέχθηκε για την υλοποίηση του παρόντος συστήματος θα γίνει στο κεφάλαιο 4) η πλήρης λειτουργία του κυκλώματος περιγράφεται σε κώδικα HDL. Τα σήματα που εισάγονται στα κυκλώματα αναπαρίστανται ως μεταβλητές και οι λογικές συναρτήσεις εκφράζονται ως αναθέσεις τιμών σε αυτές.

Βασικό πλεονέκτημα των μεθόδων HDL είναι η δυνατότητα δημιουργίας βιβλιοθηκών με τμήματα κώδικα που μπορεί να επαναχρησιμοποιηθεί - ως έχει ή με τροποποιήσεις για εισαγωγή νέων λειτουργιών - βοηθώντας την ταχύτερη ανάπτυξη νέων συστημάτων (ανάπτυξη RAD - Rapid Application Development).

Οι γλώσσες HDL είναι κατάλληλες τόσο για μικρά όσο και για μεγάλα και εξαιρετικά πολύπλοκα συστήματα. Είναι χαρακτηριστικό ότι η VHDL έχει χρησιμοποιηθεί για την περιγραφή μικροεπεξεργαστών που διαθέτουν εκατομμύρια transistor.

Αρχική Σύνθεση (Initial Synthesis)

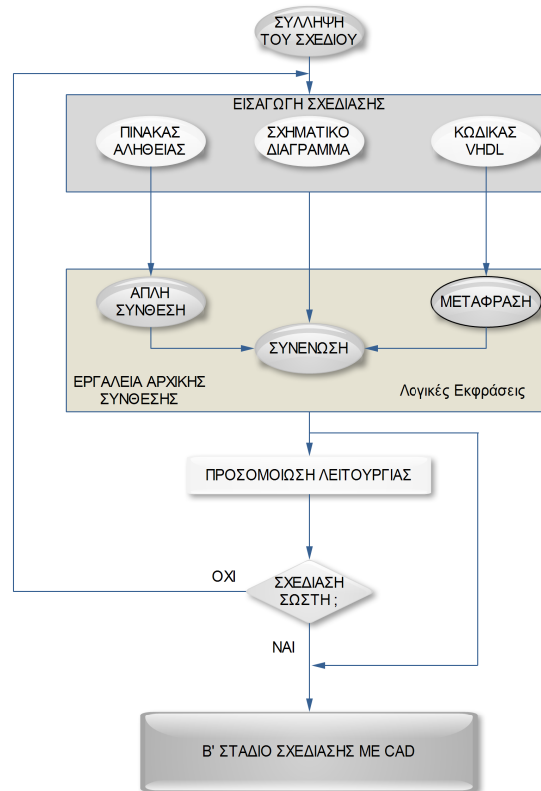
Τα συστήματα CAD μεταφράζουν (compile) τον κώδικα HDL σε ένα κύκλωμα λογικών πυλών. Αυτή είναι μια περιγραφή χαμηλού επιπέδου (low level) του κυκλώματος, σε αντιδιαστολή με τη high-level περιγραφή του κώδικα HDL.

Εάν η εισαγωγή σχεδίασης είναι με τη χρήση σχηματικού διαγράμματος τότε παράγεται ένα σύνολο λογικών εξισώσεων που αναπαριστούν το κύκλωμα του διαγράμματος. Τέλος αν χρησιμοποιείται ο πίνακας αληθείας, το εργαλείο δημιουργεί εκφράσεις για τις λογικές συναρτήσεις που είναι κωδικοποιημένες σ'αυτό τον πίνακα.

Σε κάθε περίπτωση οι λογικές εκφράσεις που παράγουν τα εργαλεία αρχικής σύνθεσης απέχουν πολύ από τη βέλτιστη δυνατή μορφή. Αυτό είναι λογική απόρροια του γεγονότος ότι ο σχεδιαστής είναι αδύνατο να συνθέσει με το χέρι το βέλτιστο κύκλωμα. Εξ'αλλου για τη λειτουργική προσομοίωση (το αμέσως επόμενο στάδιο), αυτό δεν είναι απαραίτητο. Τη βελτιστοποίηση αυτή τη διεξάγει το στάδιο της λογικής σύνθεσης που περιγράφεται παρακάτω.

Λειτουργική Προσομοίωση (Functional Simulation)

Είναι το στάδιο όπου επαληθεύεται η ορθή (ή μη) λειτουργία του υπό ανάπτυξη συστήματος. Ο χρήστης ορίζει δυνατές τιμές εισόδων και ο προσομοιωτής προσδιορίζει τις τιμές εξόδου. Η έξοδος δίνεται συνήθως με τη μορφή ενός διαγράμματος χρονισμού, με βάση το οποίο ο χρήστης αποφαινεται για την ορθότητα της λειτουργίας του συστήματος. Ένα σημείο το οποίο χρήζει προσοχής είναι ότι ο λειτουργικός προσομοιωτής θεωρεί το χρόνο διάδοσης των σημάτων μέσα από τις λογικές πύλες αμελητέο. Αυτό, προφανώς, δε συμβαίνει στο πραγματικό κύκλωμα αλλά, για τις ανάγκες της λειτουργικής επαλήθευσης, δεν είναι απαραίτητη η ρεαλιστική προσέγγιση οσον αφορά στο χρονισμό. Το διάγραμμα 1.6 απεικονίζει το πρώτο στάδιο ανάπτυξης ενός ψηφιακού συστήματος με χρήση CAD. Ακολουθεί (§1.2.5) το δεύτερο στάδιο ανάπτυξης όπου, πλέον, διενεργείται η βελτιστοποίηση των λογικών εκφράσεων του κυκλώματος και λαμβάνεται, για πρώτη φορά, υπόψη η επιθυμητή τεχνολογία υλοποίησης (π.χ. PLD). Ακριβώς για αυτό το λόγο, το επόμενο τμήμα (§1.2.4) είναι αφιερωμένο σε μια περιγραφή των τεχνολογιών υλοποίησης



Σχήμα 1.6: Α' Στάδιο Σχεδίασης CAD

1.2.4 Τεχνολογίες Υλοποίησης

Στη βάση των πάντων, αποτελώντας το πιο χαμηλό επίπεδο περιγραφής, κάθε πληροφορία είναι κωδικοποιημένη σε *bit* (δυναμικά ψηφία), τα οποία μπορούν να πάρουν τις τιμές '0' και '1'. Θεωρώντας ένα σύστημα (θετικής) αναφοράς σε τάσεις, αυτές οι τιμές αντιστοιχίζονται σε υψηλή στάθμη (high) για το λογικό '1' και χαμηλή (low) για το λογικό '0'. Πιο συγκεκριμένα: V_{ss} είναι η χαμηλότερη στάθμη και αντιστοιχεί στη γείωση (GND), V_{dd} είναι η τάση τροφοδοσίας και αντιστοιχεί στην υψηλότερη στάθμη (τυπικές τιμές είναι τα $5.5V$ και τα $3.3V$). Ορίζονται και οι ενδιάμεσες στάθμες $V_{1,min}$ και $V_{0,max}$ κατά τέτοιο τρόπο ώστε, για τα λογικά '0' και '1' να ισχύει

$$V_{ss} \leq "0" \leq V_{0,max}$$

$$V_{1,min} \leq "1" \leq V_{dd}$$

Υπάρχει, όπως φαίνεται και μια αόριστη λογική κατάσταση ('U') η οποία, εν γένει, δεν εμφανίζεται στα σήματα εκτός από τις περιπτώσεις μετάβασης από τη μία λογική κατάσταση στην άλλη ('0' ↔ '1').

$$V_{0,max} \leq "U" \leq V_{1,min}$$

Τώρα, τα λογικά κυκλώματα υλοποιούνται με transistor. Σε αυτό το επίπεδο περιγραφής είναι αρκετό να θεωρηθεί ότι τα transistor φέρονται ως διακόπτες που ελέγχονται από σήματα. Όταν η στάθμη ενός σήματος είναι χαμηλή, τότε ο διακόπτης είναι ανοιχτός (OFF) ενώ, όταν η στάθμη του σήματος είναι υψηλή, ο διακόπτης είναι κλειστός (ON).³

³Για λόγους πληρότητας, αναφέρεται ότι στην ψηφιακή τεχνολογία, τα transistor υλοποιούνται με *MOSFET* (Metal-Oxide-Semiconductor Field-Effect-Transistor). Υπάρχουν δύο τύποι MOSFET: *NMOS* με διάλυο-n και *PMOS* με διάλυο-p. Ένας συνδυα-

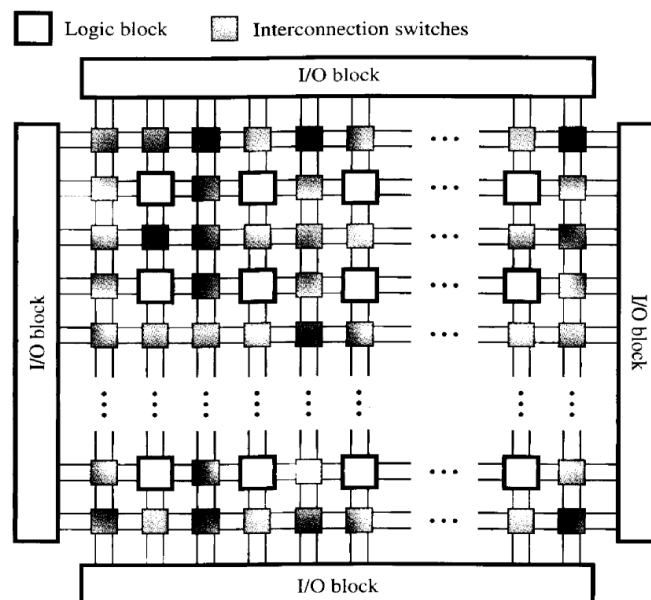
Ένα σημείο το οποίο χρήζει αναφοράς στο παρόν σημείο είναι η ταχύτητα των κυκλωμάτων λογικών πυλών. Τα transistor δεν είναι ιδανικοί διακόπτες και εμφανίζουν σημαντική αντίσταση όταν είναι ενεργά. Επιπλέον παρουσιάζουν παρασιτικές χωρητικότητες ως παραπροϊόντα της διαδικασίας κατασκευής τους. Αυτοί οι παράγοντες εισάγουν καθυστέρηση αυξάνοντας, έτσι, το χρόνο που απαιτείται για να διαδοθούν οι τιμές μέσα από τις λογικές πύλες.

FPGA

Σε αυτό το σημείο θα γίνει μια κάπως λεπτομερέστερη περιγραφή του FPGA μιας και αποτελεί την τεχνολογία υλοποίησης του παρόντος πονήματος.

Οι Λογικές Διατάξεις Πυλών Προγραμματιζόμενου Πεδίου (FPGA εφεξής) είναι ιδανικές για υλοποίηση μεγάλων κυκλωμάτων. Σε αντίθεση με άλλα ολοκληρωμένα τύπου PLD δεν περιέχουν πύλες AND και OR αλλά λογικές βαθμίδες (logical units) ως στοιχεία για την υλοποίηση λογικών εκφράσεων.

Ακολουθεί η γενική δομή του FPGA (σχήμα 1.7) : Όπως φαίνεται και στο διάγραμμα η γενική δομή μιας



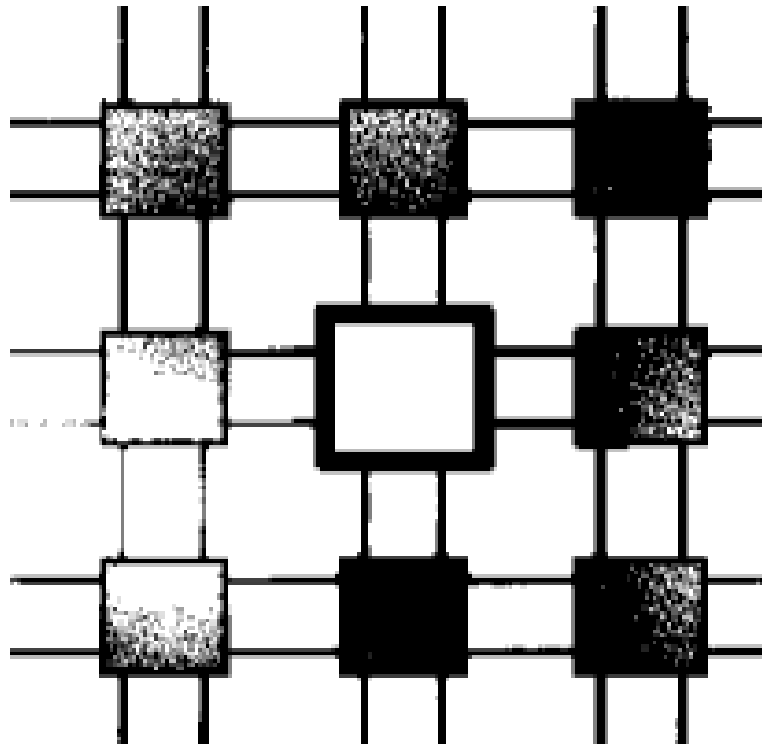
Σχήμα 1.7: Γενική Δομή του FPGA

διάταξης FPGA περιέχει τριών ειδών πόρους (resources):

1. Λογικές Βαθμίδες
2. Βαθμίδες Εισόδου/Εξόδου (I/O Units) για σύνδεση με τους ακροδέκτες της συσκευασίας
3. Προγραμματιζόμενοι Διακόπτες & Γραμμές Εσωτερικής Διασύνδεσης (Switches & Wiring)

Οι λογικές βαθμίδες σχηματίζουν έναν διδισδιάστατο πίνακα και ανάμεσα στις γραμμές και στήλες των λογικών βαθμίδων βρίσκονται τα *κανάλια δρομολόγησης* (routing channels). Τα κανάλια δρομολόγησης περιέχουν καλώδια και προγραμματιζόμενους διακόπτες και κάνουν εφικτή τη διασύνδεση των λογικών βαθμίδων με πολλούς τρόπους. Παρατηρώντας προσεκτικότερα το διάγραμμα γενικής δομής του FPGA (σχήμα 1.8) εξάγεται το συμπέρασμα ότι υπάρχουν δύο θέσεις προγραμματιζόμενων διακοπών. Τα τετράγωνα που είναι δίπλα στις λογικές βαθμίδες περιέχουν διακόπτες που συνδέουν τους ακροδέκτες I/O

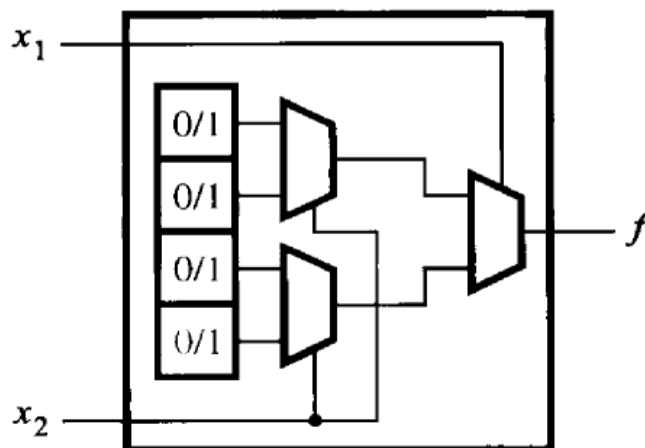
σμός των παραπάνω τύπων MOSFET αναφέρεται ως CMOS (Complementary MOS) και αποτελεί την κύρια τεχνολογία MOSFET που χρησιμοποιείται σήμερα.



Σχήμα 1.8: Λεπτομέρεια Δομής FPGA

των λογικών βαθμίδων με τα καλώδια διασύνδεσης (1η Θέση). Τα τετράγωνα που βρίσκονται διαγώνια μεταξύ των λογικών βαθμίδων συνδέουν ένα καλώδιο διασύνδεσης με ένα άλλο (π.χ. ένα οριζόντιο και ένα κατακόρυφο). Προγραμματιζόμενες συνδέσεις υπάρχουν και ανάμεσα στις βαθμίδες I/O και τα καλώδια διασύνδεσης. Ανάλογα με το μοντέλο FPGA ποικίλει και ο αριθμός των διακοπών και των καλωδίων, καθώς και το είδος των λογικών βαθμίδων.

Η συνηθέστερη μορφή λογικής βαθμίδας είναι ο Πίνακας Αναζήτησης (Look-Up Table, εφεξής LUT). Ο LUT με τη σειρά του περιέχει έναν αριθμό από κυψέλες καθεμία από τις οποίες κρατάει μια λογική τιμή 0 ή 1. Οι κυψέλες αυτές ονομάζονται *κυψέλες αποθήκευσης*. Στο σχήμα 1.9 εμφανίζεται ένας μικρός πίνακας LUT. Ο LUT του συγκεκριμένου παραδείγματος έχει δύο εισόδους και μια έξοδο, και μπορεί να υλοποιήσει



Σχήμα 1.9: Πίνακας LUT

οποιαδήποτε λογική συνάρτηση δυο μεταβλητών. Ο πλήρης πίνακας αληθείας δύο μεταβλητών περιέχει

τέσσερις γραμμές και, επομένως, ο LUT θα περιέχει τέσσερις κυψέλες αποθήκευσης, με την καθεμία από αυτές να αντιστοιχίζεται σε κάποια γραμμή του πίνακα. Οι είσοδοι του LUT αποτελούν τις εισόδους επιλογής των τριών πολυπλεκτών (MUX) οι οποίοι ανάλογα με τις τιμές των εισόδων επιλέγουν κάποια από τις τέσσερις κυψέλες και εμφανίζουν το περιεχόμενό της (0 ή 1) στην έξοδο του LUT. Αυτή είναι και η αποτίμηση της λογικής έκφρασης που αναπαρίσταται στο LUT.

Το εργαλείο σχεδίασης CAD αυτόματα μετασχηματίζει το κύκλωμα που έχει δημιουργήσει ο σχεδιαστής, ώστε να μπορεί κάθε λογική συνάρτηση του κυκλώματος να "χωρά" μέσα σε μία (και μόνο μία) λογική βαθμίδα (LUT). Στα FPGA του εμπορίου, οι πίνακες LUT έχουν, συνήθως, 4 ή 5 εισόδους οπότε και, αντίστοιχα, 16 ή 32 κυψέλες.

Κατά την υλοποίηση ενός ψηφιακού συστήματος σε FPGA, οι μεν LUT προγραμματίζονται για να υλοποιήσουν τις απαιτούμενες λογικές εκφράσεις (συναρτήσεις), τα δε κανάλια δρομολόγησης για να εκτελέσουν τις απαραίτητες διασυνδέσεις ανάμεσα στα LUT.

Ένα σημείο που οφείλει να τονιστεί εδώ, είναι ότι, στην παραγωγικότητα, η λογική βαθμίδα ενός FPGA περιλαμβάνει και άλλα κυκλώματα, όπως φαίνεται και στο σχήμα 1.10 (Μοντέλο Altera FLEX 10K).

όμως η περιγραφή τους διαφεύγει του σκοπού του παρόντος πονήματος.

Όσον αφορά στη διασύνδεση του FPGA με τη μονάδα προγραμματισμού, υποστηρίζεται η τεχνική ISP. Σύμφωνα με αυτή, στη μητρική πλακέτα (PCB), όπου τοποθετείται το FPGA, περιλαμβάνεται ένας συνδετήρας (connector) και ένα καλώδιο που τον συνδέει με τον υπολογιστή. Ο προγραμματισμός του FPGA γίνεται με τη μεταφορά της απαιτούμενης πληροφορίας, που δημιουργείται από ένα πρόγραμμα CAD, στο FPGA μέσω του καλωδίου. Το IEEE έχει προτυποποιήσει αυτά τα καλώδια μεταφοράς με το όνομα "Θύρα JTAG" (JTAG Port - Joint Test Action Group).

Μία τελευταία παρατήρηση για τις κυψελίδες αποθήκευσης των LUT, αφορά στη *πητικότητα* τους (volatile) με την έννοια ότι τα δεδομένα που περιέχουν χάνονται κατά τη διακοπή της τροφοδοσίας. Σαν αποτέλεσμα, το FPGA πρέπει να προγραμματίζεται εκ νέου κάθε φορά που τροφοδοτείται. Συχνά, για την αποφυγή της παραπάνω διαδικασίας, στο PCB, μαζί με το FPGA, περιλαμβάνεται και μια ολοκληρωμένη μνήμη ROM (Programmable ROM) που διατηρεί επί μόνιμου βάσεως τα δεδομένα της διάταξης. Τα δεδομένα της μεταφέρονται αυτόματα στις κυψέλες των LUT του FPGA.

1.2.5 Δομικά Στοιχεία Συστημάτων CAD - Μέρος Β

Λογική Σύνθεση & Βελτιστοποίηση (Logical Synthesis & Optimization)

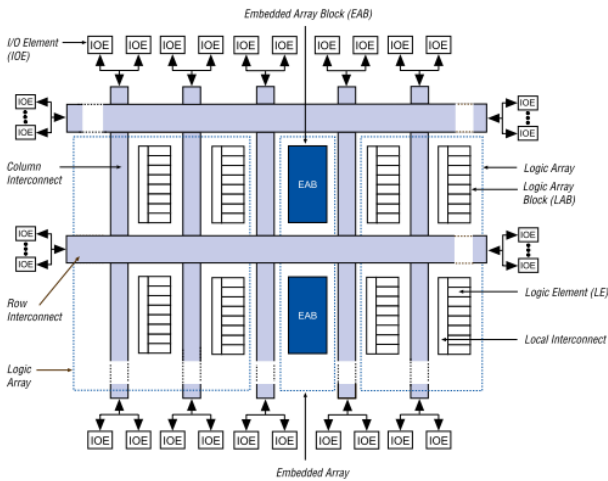
Σε αυτό το σημείο είναι χρήσιμο (για λόγους πληρότητας) να γίνει μια κατάταξη των εργαλείων λογικής σύνθεσης των συστημάτων CAD σε δύο κατηγορίες, ανάλογα με τη σχέση τους με την τεχνολογία υλοποίησης του τελικού ολοκληρωμένου:

- *Λογική Σύνθεση Ανεξαρτητη Τεχνολογίας (Technology Independent LS)
- *Λογική Σύνθεση με Χαρτογράφηση Τεχνολογίας (Technology Mapping)

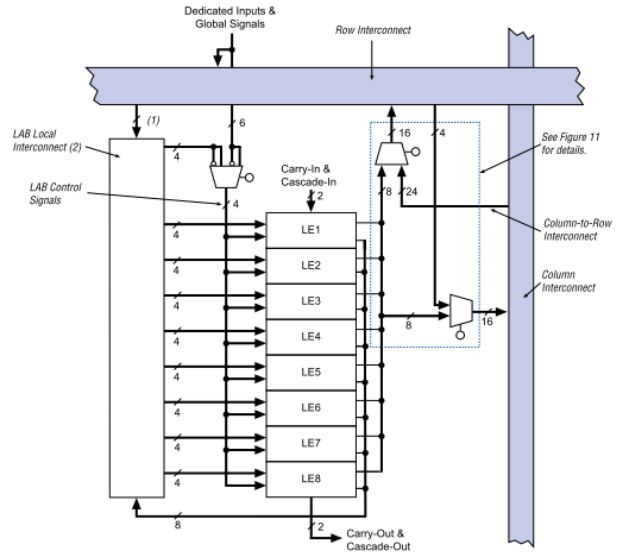
Η 1η κατηγορία αναφέρεται σε τεχνικές βελτιστοποίησης που δε λαμβάνουν υπόψη τους απαιτούμενους πόρους της πλατφόρμας υλοποίησης του κυκλώματος. Η 2η κατηγορία αναφέρεται στις τεχνικές που διασφαλίζουν ότι το τελικό κύκλωμα που θα παραχθεί μετά τη λογική σύνθεση, δύναται να υλοποιηθεί με τους πόρους που είναι διαθέσιμοι στην πλατφόρμα ανάπτυξης. Τα περισσότερα επαγγελματικά συστήματα χρησιμοποιούν και τις δύο τεχνικές.

Τώρα, η λογική σύνθεση είναι ένα στάδιο που εκτελείται αυτόματα από το CAD. Το αποτέλεσμα της απεικονίζεται σε ένα *αρχείο αναφοράς* (report file) που περιέχει το σύνολο των βελτιστοποιημένων λογικών εκφράσεων που περιγράφουν το συντιθέμενο σύστημα.

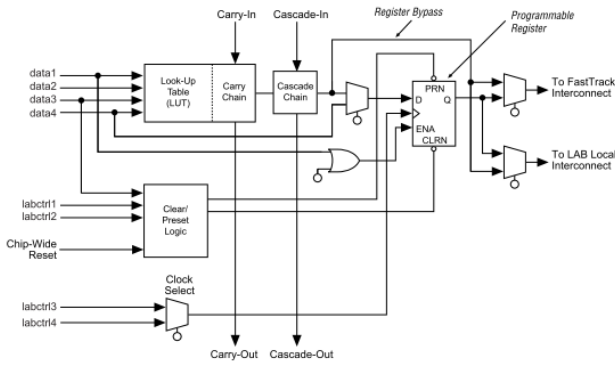
Το είδος της βελτιστοποίησης του κυκλώματος εξαρτάται από την τεχνολογία υλοποίησης, στα επαγγελ-



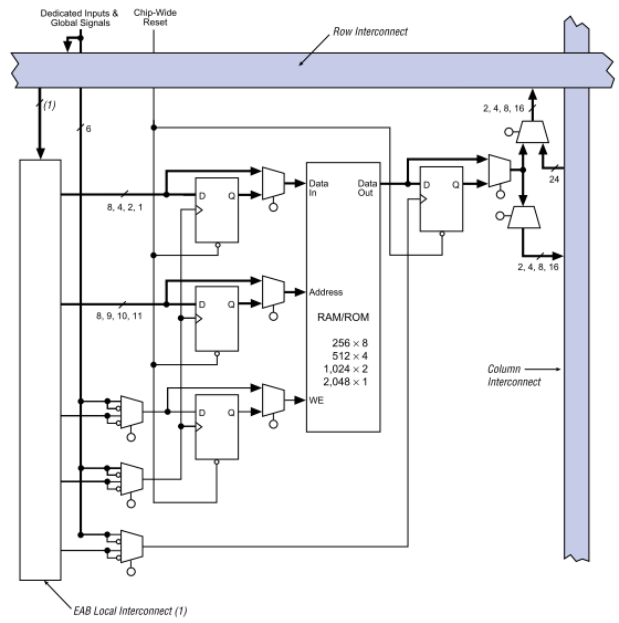
(α) Λογική Διάταξη



(β) Βαθμίδα Λογικής Διάταξης - LAB

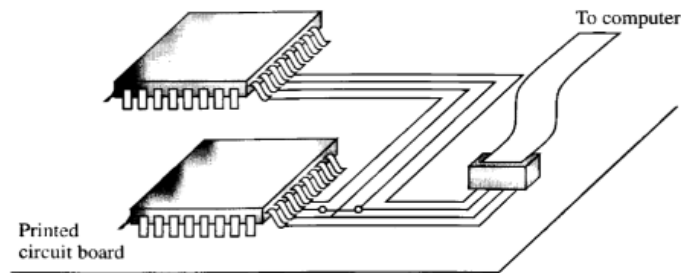


(γ) Λογικό Στοιχείο



(δ) Ενσωματωμένη Βαθμίδα - EAB

Σχήμα 1.10: Παραδείγματα Λογικών Βαθμίδων Εμπορικών Συστημάτων FPGA



Σχήμα 1.11: Διασύνδεση JTAG

ματικά συστήματα CAD. Παραδείγματος χάρη για το FPGA, που περιέχει LUT, ο αριθμός των εισόδων ενός λογικού μπλοκ καθορίζεται ευθέως από το μέγεθος του LUT του συγκεκριμένου μοντέλου FPGA.

Πρακτικά Ζητήματα - Ελαχιστοποίηση Λογικών Εκφράσεων σε Συστήματα CAD

Πέρα από τις γνωστές μεθόδους παράστασης λογικών συναρτήσεων με χρήση πινάκων αληθείας, εκφράσεων άλγεβρας Boole, διαγραμμάτων Venn και πινάκων Karnaugh, υπάρχει και η δυνατότητα απεικόνισης των n μεταβλητών μιας λογικής έκφρασης σε ένα κύβο n διαστάσεων.

Η τελευταία μορφή απεικόνισης είναι κατάλληλη για τη δημιουργία αλγορίθμων ελαχιστοποίησης, κατάλληλων για εκτέλεση από υπολογιστή. Επιγραμματικά⁴ αναφέρονται κάποιοι από αυτούς, που χρησιμοποιούνται στα σύγχρονα συστήματα CAD).

- Γενίκευση των Πρώτων Όρων Αληθείας (Prime Implicants)
- Πράξη Γινομένου Αστέρος '*'
- Προσδιορισμός των Prime Implicants με χρήση της Πράξης Αποκοπής '#'
- Προσδιορισμός Ελάχιστης Καλυψης με Ευριστική Προσέγγιση Διακλάδωσης (Branching Heuristic Approach)

Φυσική Χωροθέτηση (Layout Synthesis)

Αυτόματο βήμα στο οποίο γίνεται ο ακριβής προσδιορισμός της φυσικής χωροθέτησης του συστήματος με βάση την τεχνολογία υλοποίησης που επιλέχθηκε (target technology). Συνίσταται από δύο κύριες διαδικασίες:⁵

- Τοποθέτηση (Placement) :: Καθορίζει επακριβώς σε ποιο σημείο του IC θα υλοποιηθεί έκαστο βελτιστοποιημένο λογικό μπλοκ
- Δρομολόγηση (Routing) :: Επιλογή καλωδιώσεων του IC που θα χρησιμοποιηθούν για τις απαιτούμενες διασυνδέσεις.

Και οι δύο προαναφερθείσες διαδικασίες εμφανίζουν ισχυρή εξάρτηση από την τεχνολογία υλοποίησης. Για παράδειγμα, σε ένα FPGA, στο στάδιο της τοποθέτησης κάθε λογική συνάρτηση αντιστοιχίζεται σε μια λογική κυψέλη. Στο στάδιο της δρομολόγησης χρησιμοποιούνται οι επαναπρογραμματιζόμενοι διακόπτες για τη σύνδεση των λογικών κυψελών μεταξύ τους.

Αυτό το στάδιο είναι και το πιο απαιτητικό για τα εργαλεία CAD, με τη δυσκολία να αυξάνει εκθετικά με την πολυπλοκότητα του υπο υλοποίηση ψηφιακού συστήματος. Οι αλγόριθμοι φυσικής χωροθέτησης αποτελούν διαρκές αντικείμενο μεγάλης ερευνητικής δραστηριότητας.

Προσομοίωση Χρονισμού (Timing Simulation)

Στην προσομοίωση χρονισμού εμφανίζονται, πλέον, οι πραγματικές καθυστερήσεις διάδοσης των σημάτων μέσα από τις λογικές πύλες. Οι καθυστερήσεις αυτές εξαρτώνται άμεσα από την τεχνολογία υλοποίησης. Το μοντέλο που παράγεται από το CAD εγκολπώνει τις καθυστερήσεις που εμφανίζονται στις λογικές κυψέλες (FPGA) και στην καλωδίωση διασύνδεσης (wiring). Σε περίπτωση προβλημάτων, η επαναχρησιμοποίηση του εργαλείου λογικής σύνθεσης θα επιφέρει, ενδεχομένως (με την υιοθέτηση διαφόρων

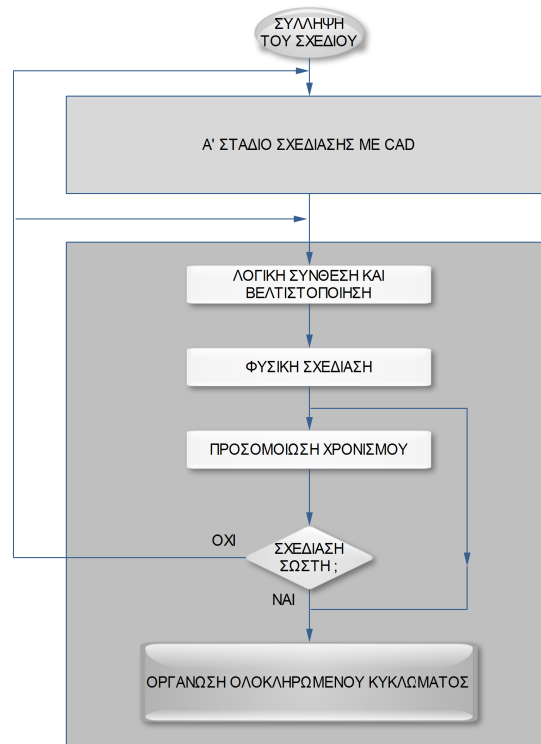
⁴Ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στο [BV01, Κεφ. 4]

⁵Συνήθως αναφέρονται ομού ως *place n' route*

παραμέτρων βελτιστοποίησης), τα επιθυμητά αποτελέσματα. Σε καθε άλλη περίπτωση απαιτείται επανασχεδιασμός μέρους ή όλου του συστήματος.

Σε αυτό το σημείο είναι χρήσιμο να τονιστεί το βασικό trade-off που προκύπτει κατά τη σχεδίαση ψηφιακών κυκλωμάτων : "Χώρος ή Ταχύτητα". Συγκεκριμένα, ο χώρος (και εντέλει το κόστος) βελτιστοποιείται με την ελαχιστοποίηση της επιφάνειας του IC που απαιτείται για την υλοποίηση του συστήματος. Αυτή είναι μια κρίσιμη παράμετρος, ιδιαίτερα σε περιπτώσεις όπου η ανάπτυξη γίνεται σε συγκεκριμένο μοντέλο PLD ή όταν η φύση της εφαρμογής απαιτεί τελικό κύκλωμα μικρών διαστάσεων. Από την άλλη, η ταχύτητα βελτιστοποιείται όταν ελαχιστοποιούνται οι χρονικές καθυστερήσεις που επιβάλλει το σύστημα (είτε λόγω σχεδίασης είτε λόγω φυσικής) στα σήματα, πολλές φορές με κόστος την αυξημένη επιφάνεια του κυκλώματος. Ένα παράδειγμα εφαρμογών όπου η ταχύτητα είναι ζωτικός παράγοντας επιτυχίας είναι τα real-time συστήματα. Σε καθε περίπτωση, η κατάληξη είναι η αποτύπωση του ψηφιακού συστήματος στο ολοκληρωμένο (target IC).

Το διάγραμμα ροής του σχήματος 1.12 απεικονίζει το β' στάδιο της διαδικασίας σχεδίασης με CAD.



Σχήμα 1.12: Β' Στάδιο Σχεδίασης CAD

1.2.6 Ένα Διαφωτιστικό Παράδειγμα

Για λόγους κατανόησης παρατίθεται ένα πλήρες παράδειγμα σχεδίασης ενός απλού ψηφιακού κυκλώματος, το οποίο περιλαμβάνει το α' στάδιο σχεδίασης με CAD καθώς και την τελική μορφή του προγραμματισμένου FPGA.

Έστω, λοιπόν, ότι η λογική έκφραση που χαρακτηρίζει τη λειτουργία του υπο σχεδίαση κυκλώματος είναι η :

$$F = x_1x_2\bar{x}_3 + x_1x_2x_3 + \bar{x}_2x_3x_1 + \bar{x}_2x_3\bar{x}_1$$

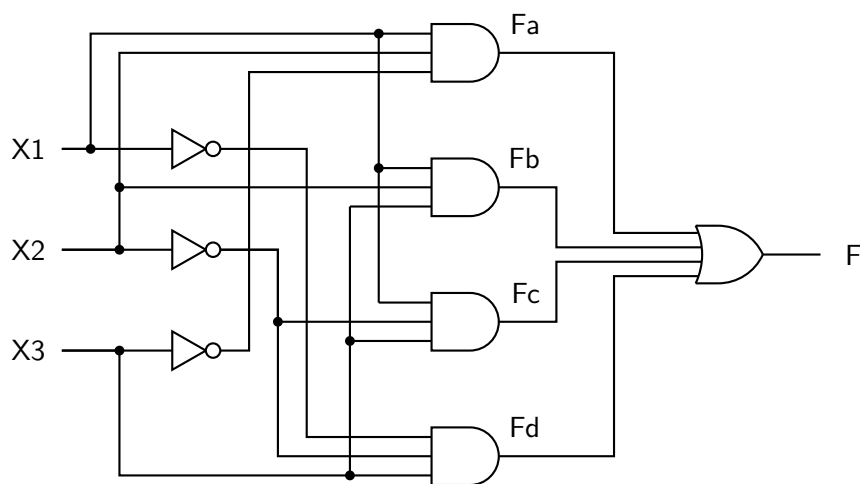
Πίνακας Αληθείας

Ο πίνακας αληθείας που ακολουθεί αποτυπώνει τις τιμές της εξόδου του κυκλώματος για όλους τους δυνατούς συνδυασμούς των εισόδων.

x_1	x_2	x_3	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Γράφημα Πυλών

Το Σχήμα 1.13 απεικονίζει το λογικό κύκλωμα ως διάγραμμα πυλών. Η πολυπλοκότητά του, σε αυτό το στάδιο (όπου δεν έχει υπάρξει ακόμη κάποια βελτιστοποίηση), είναι : 3 πύλες NOT, 4 πύλες AND των 3 εισόδων και 1 πύλη OR των 4 εισόδων.



Σχήμα 1.13: Αρχικό Λογικό Κύκλωμα

Κώδικας VHDL

Ο κώδικας VHDL που εισάγεται στο σύστημα CAD ως περιγραφή του κυκλώματος ακολουθεί αμέσως παρακάτω. Ο κώδικας αυτός αναπαριστάται στην περιγραφή του κυκλώματος όπως αυτό εμφανίζεται στο Σχήμα 1.13. Είναι ακριβώς η μορφή της προδιαγραφής του κυκλώματος από το σχεδιαστή, χωρίς βελτιστοποίηση.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity circ_1 is
  port(
    clk,rst : in std_logic;
    --Input
    x1 : in bit;
```

```

    x2 : in bit;
    x3 : in bit;
    --Output
    f : out bit
  );
end circ_1;

architecture ca of circ_1 is
  signal x1_bar, x2_bar, x3_bar : bit;
  signal fa, fb, fc, fd : bit;
begin
  x1_bar <= not x1;
  x2_bar <= not x2;
  x3_bar <= not x3;
  fa <= x1 and x2 and x3_bar;
  fb <= x1 and x2 and x3;
  fc <= x1 and x2_bar and x3;
  fd <= x1_bar and x2_bar and x3;
  f <= fa or fb or fc or fd;
end ca;

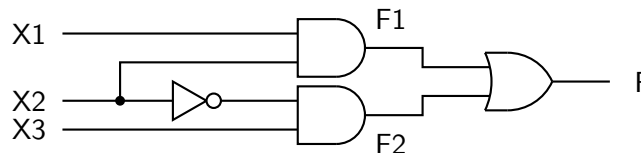
```

Αρχική Σύνθεση - Βελτιστοποίηση Λογικής Έκφρασης

Σε αυτό το στάδιο, της αρχικής σύνθεσης, είναι που πραγματοποιείται η απλοποίηση της λογικής συνάρτησης από το εργαλείο CAD.⁶ Η βελτιστοποιημένη λογική έκφραση έχει ως εξής :

$$F = F_{opt} = x_1x_2 + \bar{x}_2x_3 = f_1 + f_2$$

Στο Σχήμα 1.14 απεικονίζεται το διάγραμμα λογικών πυλών του απλοποιημένου κυκλώματος. Η λειτουργία του είναι ακριβώς ίδια με αυτή του κυκλώματος του Σχήματος 1.13 και είναι φανερό ότι η πολυπλοκότητα έχει μειωθεί αισθητά : 1 πύλη NOT, 2 πύλες AND των 2 εισόδων και 1 πύλη OR των 2 εισόδων.



Σχήμα 1.14: Βελτιστοποιημένο Λογικό Κύκλωμα

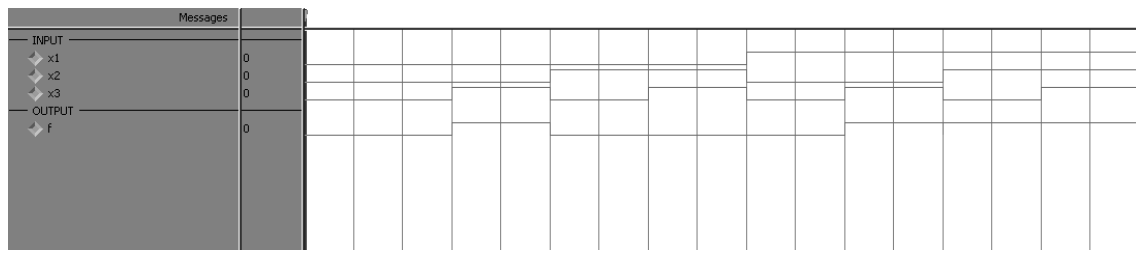
Λειτουργική Προσομοίωση - Διάγραμμα Κυματομορφών

Στο Σχήμα 1.15 απεικονίζεται το διάγραμμα κυματομορφών, όπως αυτό παράγεται από το εργαλείο του συστήματος CAD, που επιβεβαιώνει την ορθή λειτουργία του κυκλώματος. Είναι ένα βασικό εργαλείο καθώς με αυτό, ο σχεδιαστής μπορεί να εντοπίσει και διορθώσει ατέλειες και λάθη στην αρχιτεκτονική του συστήματος που υλοποιεί.

Πίνακες LUT του FPGA

Στο Σχήμα 1.16 εμφανίζονται οι πίνακες LUT μιας προγραμματισμένης βαθμίδας του FPGA, μετά από τη διαδικασία της λογικής σύνθεσης. Αποτελούν τη φυσική υλοποίηση της βελτιστοποιημένης λογικής έκφρασης στο τελικό κύκλωμα.

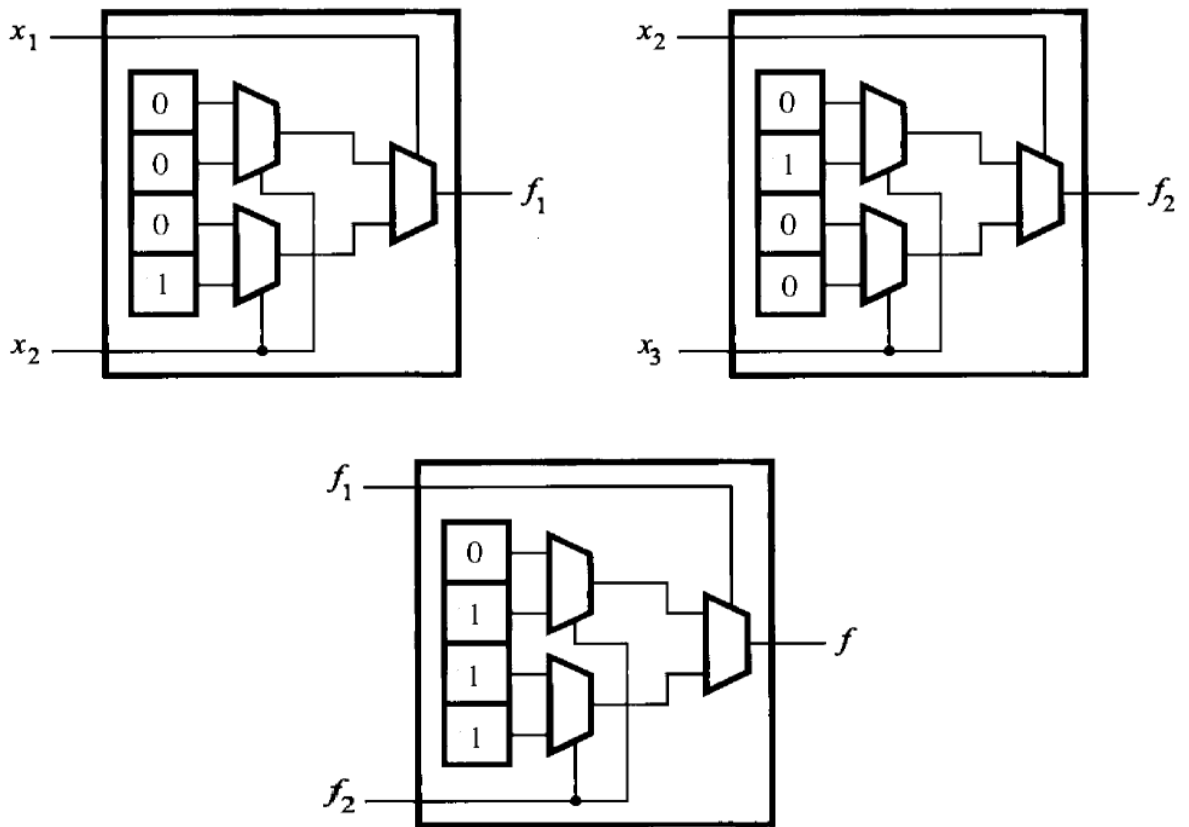
⁶Το συγκεκριμένο παράδειγμα, φυσικά, είναι πολύ απλό και στην πραγματικότητα, μπορεί εύκολα να βελτιστοποιηθεί με το "χέρι" με απλή εφαρμογή των νόμων *DeMorgan*.



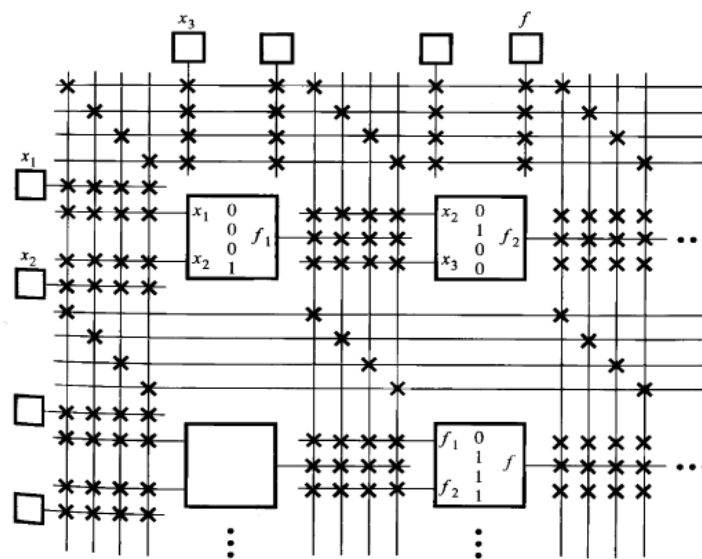
Σχήμα 1.15: Κυματομορφές Λειτουργικής Προσομοίωσης

Προγραμματισμένο FPGA

Στο Σχήμα 1.17 εμφανίζονται οι λογικές βαθμίδες που εμπλέκονται στην υλοποίηση του κυκλώματος και οι αντίστοιχες διασυνδέσεις.



Σχήμα 1.16: Προγραμματισμένοι LUT



Σχήμα 1.17: Τμήμα του προγραμματισμένου FPGA

Κεφάλαιο 2

Τηλεπικοινωνιακές προδιαγραφές

2.1 Μέρος Α - Θεωρητική ανάλυση συστήματος OFDM

2.1.1 Γενικά

Η διαμόρφωση OFDM (Ορθογωνική πολυπλεξία με διαίρεση συχνότητας) ανήκει στην κατηγορία των ζωνοπερατών διαμορφώσεων με πολλαπλά φέροντα ¹ (multiple carrier bandpass modulation) και χαρακτηρίζεται ως ευρυζωνική διαμόρφωση (wideband modulation). Αυτή η τεχνική μετάδοσης, επιτυγχάνει πολύ αποδοτική χρησιμοποίηση του διαθέσιμου εύρους ζώνης σε ένα κανάλι, παρέχοντας υψηλούς ρυθμούς μετάδοσης δεδομένων. Επιπλέον το OFDM παρουσιάζει αυξημένη ανθεκτικότητα σε κανάλια με πολυδιαδρομική διάδοση (multipath propagation).

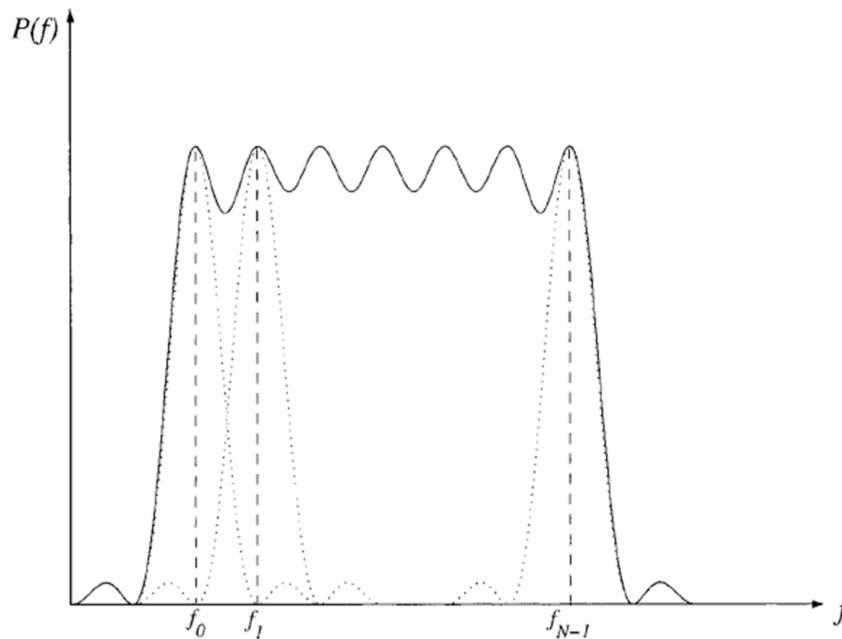
2.1.2 Αρχές λειτουργίας

Η βασική αρχή του OFDM είναι η διαίρεση του διαθέσιμου φάσματος του καναλιού ευρείας ζώνης, σε πολλά υποκανάλια (subchannels) στενής ζώνης. Η προς μετάδοση πληροφορία σπάει σε μικρά κομμάτια, τα οποία μεταδίδονται παράλληλα μέσα από τα υποκανάλια, έχοντας διαμορφωθεί με έναν κλασικό τύπο διαμόρφωσης ενός φέροντος (όπως QAM, PAM, PSK). Το κάθε ένα από αυτά τα φέροντα ονομάζεται υποφέρον (subcarrier). Δηλαδή με κάθε OFDM σύμβολο μεταδίδεται ένα τμήμα (chunk) πληροφορίας. Με άλλα λόγια, ένα σύστημα OFDM είναι ισοδύναμο με πολλά συστήματα μιας φέρουσας, που μεταφέρουν από κοινού κομμάτια της ίδιας αρχικής πληροφορίας. Οι τελικοί παλμοί που παράγονται είναι μεγάλης χρονικής διάρκειας (και μικρού εύρους ζώνης), γεγονός που τους καθιστά ανθεκτικούς σε φαινόμενα συχνοεπιλεκτικότητας του ραδιοδιαύλου (που οφείλονται στο multipath).

Επιπλέον, στο OFDM, τα φάσματα των υποκαναλιών είναι επικαλυπτόμενα (στη συχνότητα), πλην όμως ορθογώνια μεταξύ τους. Έτσι είναι δυνατή η πολύ αποδοτική εκμετάλλευση του εύρους ζώνης, ενώ η ορθογωνιότητα εξασφαλίζει ότι στα σημεία όπου το φάσμα ενός υποκαναλιού παρουσιάζει μέγιστο, το φάσμα των γειτονικών υποκαναλιών παρουσιάζει μηδενισμό (Σχήμα 2.1).

Ένα σημείο δυσκολίας κατά την προσπάθεια υλοποίησης ενός συστήματος OFDM, είναι η αναγκαιότητα χρήσης πολλών ταλαντωτών και μικτών για την διαμόρφωση των subcarrier. Ωστόσο όπως έχει αποδειχτεί από τον Weinstein το 1971, η διαμόρφωση των subcarrier μπορεί να γίνει ψηφιακά μέσω του Διακριτού Μετασχηματισμού Φουριέ (Discrete Fourier Transform - DFT). Στις σημερινές παρακτικές υλοποιήσεις, χρησιμοποιείται ο FFT (Fast Fourier Transform), ένας αλγόριθμος υπολογισμού των DFT και IDFT. Μεγάλο του πλεονέκτημα αποτελεί η μικρή του πολυπλοκότητα και η αντίστοιχη μείωση στο απαιτούμενο υλικό. Αυτός είναι ο πυρήνας της διαμόρφωσης OFDM και χάρη στην συνεχή έρευνα και ανάπτυξη τεχνικών αποδοτικού υπολογισμού του FFT, είναι δυνατή η ευρεία εφαρμογή του OFDM.

¹Το καθένα από αυτά τα φέροντα ονομάζεται υποφέρον (sub-carrier).



Σχήμα 2.1: Φάσμα subcarrier OFDM

Σύγκριση με συστήματα μιας φέρουσας

Το OFDM παρουσιάζει αρκετά πλεονεκτήματα έναντι των διαμορφώσεων μιας φέρουσας. Η εύκολη ισοστάθμιση του καναλιού (channel equalization) είναι μία από αυτές. Με τη χρήση διαμόρφωσης OFDM, ένα συχνοεπιλεκτικό κανάλι (frequency selective channel) εκφυλίζεται σε μια συστάδα από επίπεδα κανάλια (flat channel). Σε ένα επίπεδο κανάλι, η διαδικασία της ισοστάθμισης είναι αρκετά απλή. Αντιθέτως, σε ένα συχνοεπιλεκτικό κανάλι, η ισοστάθμιση είναι μία υπολογιστικά πολύπλοκη διαδικασία, γνωστή ως το *πρόβλημα της αποσυνέλιξης*. Επιπλέον σε ένα OFDM σύστημα, υπάρχει η δυνατότητα δυναμικής προσαρμογής του τύπου διαμόρφωσης σε κάθε subcarrier, ανάλογα με την κατάστασή του, εξασφαλίζοντας βέλτιστη κατανομή της διαθέσιμης ισχύος στα υποκανάλια. Τέλος, λόγω της κυκλικής επέκτασης, για την οποία θα γίνει αναφορά παρακάτω, το OFDM είναι ανθεκτικό σε σφάλματα συγχρονισμού στο χρόνο μεταξύ πομπού και δέκτη.

Ιδιαιτερότητες ενός πραγματικού συστήματος OFDM

Οι κυριότερες προκλήσεις που έχει να αντιμετωπίσει ένας σχεδιαστής συστήματος OFDM είναι :

- A) Η αυξημένη ευαισθησία σε σφάλματα ολίσθησης συχνότητας, που οδηγούν σε διακαναλική παρεμβολή (ICI). Αυτά εμφανίζονται λόγω φαινομένων Doppler κατά τη διέλευση του σήματος από το κανάλι, ή λόγω αποκλίσεων στις συχνότητες λειτουργίας, τόσο των ταλαντωτών RF, όσο και του ζεύγους DAC/ADC, ανάμεσα στον πομπό και το δέκτη.
- B) Οι υψηλές τιμές του PAPR², οι οποίες μάλιστα, μεγαλώνουν με την αύξηση του αριθμού των subcarrier. Το κύριο πρόβλημα, εδώ, συνίσταται στο ότι οι ενισχυτές RF έχουν περιορισμένο εύρος γραμμικής λειτουργίας, έξω από το οποίο προκύπτει ψαλιδισμός του σήματος, λόγω του κόρου, ο οποίος εισάγει παραμόρφωση και προκαλεί αλλοίωση των φασματικών χαρακτηριστικών του σήματος. Η πιθανότητα εμφάνισης ψαλιδισμού είναι μεγαλύτερη καθώς αυξάνουν οι τιμές του PAPR. Προκειμένου να αποφευχθούν τέτοια φαινόμενα, οι ενισχυτές λειτουργούν σε χαμηλότερη μέση στάθμη ισχύος από τη μέγιστη που δύνανται να παράγουν, πράγμα που οδηγεί στη μειωμένη αποδοτικότητά τους.

²Peak-to-Average-Power Ratio, δηλαδή ο λόγος της μέγιστης προς τη μέση ισχύ του εκπεμπόμενου σήματος

2.1.3 Περιγραφή Λειτουργίας

Τα βασικά συστατικά μέρη ενός τηλεπικοινωνιακού συστήματος OFDM, απεικονίζονται στο σχήμα 2.2.

Παρακάτω περιγράφονται συνοπτικά τα μέρη αυτά και επεξηγούνται οι λειτουργίες που επιτελούνται τόσο κατά τη μετάδοση, όσο και κατά τη λήψη OFDM σήματος.

Πομπός

Η ακολουθία των προς μετάδοση ψηφιακών δεδομένων, μετατρέπεται σε N παράλληλες ακολουθίες από μπλοκ των k bits (k -block). Το κάθε k -block μέσα από τη διαδικασία της αντιστοίχισης αστερισμού (constellation mapping), αντιστοιχεί σε ένα μιγαδικό σύμβολο ενός αστερισμού M -αδικής διαμόρφωσης (όπως QAM, PSK, PAM), δηλαδή διαμορφώνεται σε subsymbol βασικής ζώνης. Το αποτέλεσμα είναι N παράλληλες ακολουθίες συμβόλων. Μία ομάδα από N (παράλληλα) σύμβολα τροφοδοτείται στον IFFT όπου διαμορφώνεται σε μιγαδικό σήμα βασικής ζώνης στο πεδίο του χρόνου, παράγοντας ένα ωφέλιμο OFDM σύμβολο (useful OFDM symbol). Οι δύο συνιστώσες ονομάζονται συμφασική (I) και ορθογωνική (Q), για το πραγματικό και το φανταστικό μέρος του σήματος αντίστοιχα.

Στη συνέχεια προστίθεται στο σήμα ένα πρόθεμα κυκλικής επέκτασης (Cyclic Prefix - CP), με αποτέλεσμα την παραγωγή ενός OFDM συμβόλου (OFDM symbol). Το CP είναι μία απλή επανάληψη ενός τμήματος από το τέλος του σήματος στην αρχή του. Ο ρόλος και η αναγκαιότητά του θα εξηγηθούν παρακάτω. Μέχρι αυτό το σημείο έχει παραχθεί το ψηφιακό σήμα στο πεδίο του χρόνου.³

Κατόπιν το σήμα μετατρέπεται σε αναλογικό μέσω δύο DAC και διαμορφώνεται σε IF πραγματικό ζωνοπερατό σήμα. Τέλος, υψώνεται στις RF συχνότητες (upconversion), και εκπέμπεται από την κεραία στο κανάλι.

Δέκτης

Το RF σήμα που φτάνει στην κεραία λήψης του δέκτη κατεβαίνει σε IF πραγματικό σήμα (downconversion). Στη συνέχεια, αποδιαμορφώνεται σε μιγαδικό σήμα βασικής ζώνης και ψηφιοποιείται μέσω δύο ADC.

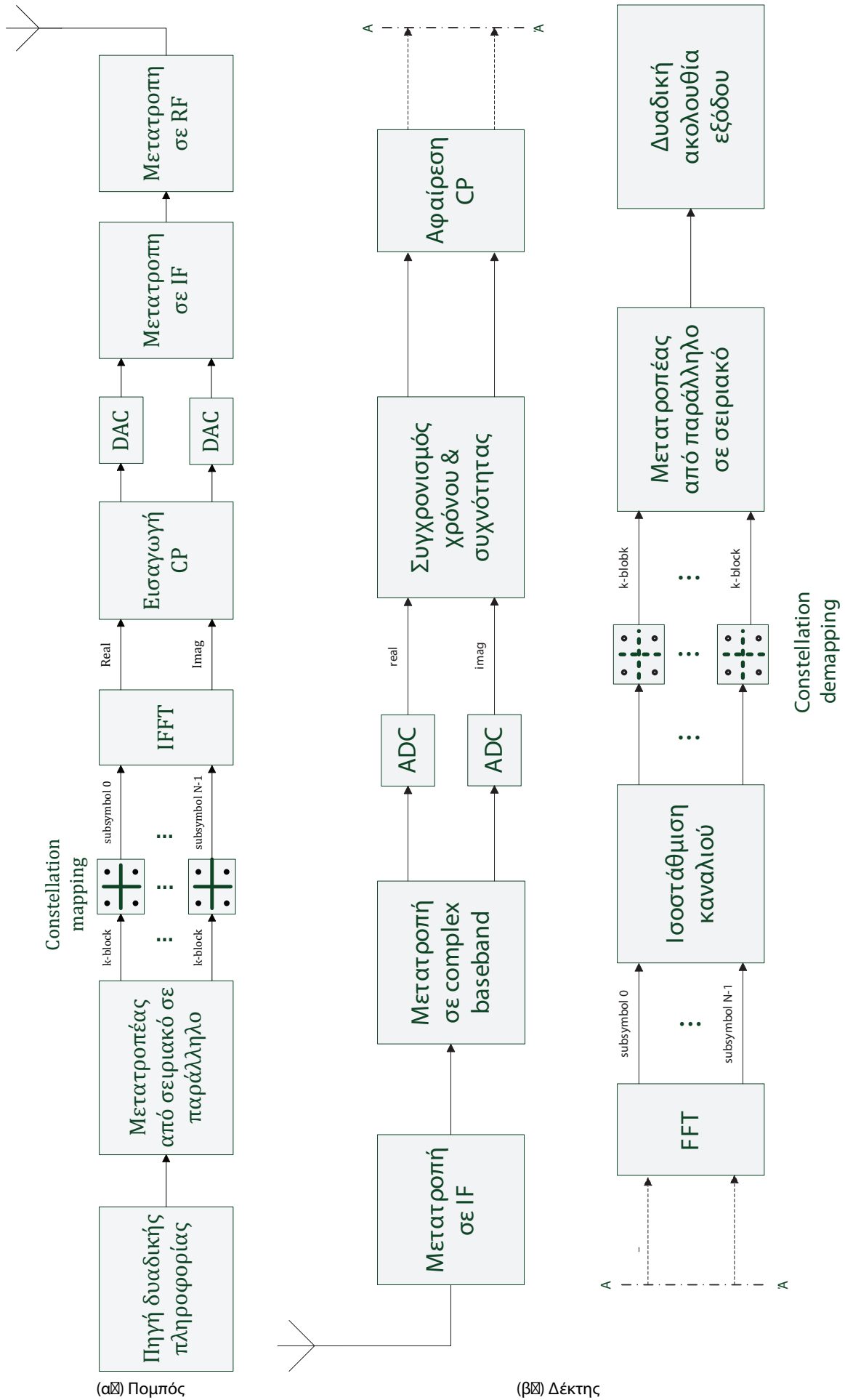
Εφόσον ανιχνευτεί η αρχή της μετάδοσης, από τη διαδικασία συγχρονισμού του δέκτη, αφαιρείται η κυκλική επέκταση και το samples του χρονικού σήματος τροφοδοτούνται στον FFT. Εκεί γίνεται η αποδιαμόρφωση βασικής ζώνης και παράγονται N παράλληλες ακολουθίες μιγαδικών συμβόλων. Τα σύμβολα αυτά είναι αλλοιωμένα από το multipath κανάλι και τον προσθετικό θόρυβο (Additive White Gaussian Noise - AWGN), σε σχέση με τα αντίστοιχα που έχουν μεταδοθεί. Σε αυτό το σημείο απομακρύνεται η επίδραση του καναλιού μέσω του σταδίου εκτίμησης και ισοστάθμισης καναλιού (channel estimation and equalization). Στη συνέχεια ακολουθεί η αποδιαμόρφωση των subsymbol (constellation demapping) και η ανακατασκευή των N παράλληλων ακολουθιών των k -block. Οι ακολουθίες αυτές, δύναται να διαφέρουν από τις αντίστοιχες που μεταδόθηκαν λόγω της αλλοίωσης των συμβόλων. Τέλος οι ακολουθίες των k -block συναρμολογούνται και πάλι στην αρχική ακολουθία των ψηφιακών δεδομένων.

2.1.4 Επεξηγήσεις

Constellation mapping

Τα σύμβολα ενός αστερισμού (οι κυματομορφές, δηλαδή, που αντιστοιχούν στα k -block), μπορούν εν γένει να αναλυθούν σε ένα ορθοκανονικό N -διάστατο σύστημα βάσης (αποτελούμενο από N γραμμικά ανεξάρτητες κυματομορφές). Δεδομένης μιας τέτοιας βάσης, το κάθε σύμβολο ορίζεται ως μια N -άδα συντεταγμένων.

³Το ψηφιακό σήμα που πρόκειται να μετατραπεί σε αναλογικό ή προέρχεται από αναλογικό, αποτελείται από τα δείγματα (samples) του συνεχούς χρονικού σήματος σε ισαπέχοντα χρονικά διαστήματα, ίσα με την περίοδο δειγματοληψίας.



(α) Πομπός

(β) Δέκτης

Σχήμα 2.2: Τηλεπικοινωνιακό μπλοκ διάγραμμα πομπού και δέκτη OFDM

Στην ειδική περίπτωση όπου $N = 2$ το κάθε σύμβολο ορίζεται ως ένα ζεύγος συντεταγμένων, έστω $s_x = (s_{x1}, s_{x2})$ ή ακόμα ως ένας μιγαδικός αριθμός $\bar{s}_x = (s_{x1} + js_{x2})$ και είναι δυνατό να απεικονιστεί γεωμετρικά στο μιγαδικό επίπεδο. Η γεωμετρική αυτή αναπαράσταση καλείται **αστερισμός** της διαμόρφωσης και είναι ένα χρήσιμο οπτικό εργαλείο για την αξιολόγηση και την κατηγοριοποίηση των διαφορετικού τύπου διαμορφώσεων. Δισδιάστατες διαμορφώσεις, μεταξύ άλλων, είναι οι QAM⁴ και PSK.

Συνθήκη ορθογωνιότητας

Η χρονική διάρκεια του ωφέλιμου συμβόλου OFDM (T_o), πρέπει να ικανοποιεί τη συνθήκη ορθογωνιότητας δηλαδή $\Delta_f = 1/T_o$, όπου με Δ_f συμβολίζουμε την απόσταση μεταξύ των subcarrier. Αν δεν ικανοποιηθεί αυτή η απαίτηση, εμφανίζεται διακαναλική παρεμβολή (ICI) και η αποδιαμόρφωση καθίσταται προβληματική.

Κυκλική επέκταση

Η κυκλική επέκταση (cyclic prefix) ή αλλιώς το διάστημα φύλαξης (guard interval) έχει δύο λειτουργίες.

Αφενός προστατεύει το σήμα από την διασυμβολική παρεμβολή (ISI), αφήνοντας ένα «κενό» διάστημα μεταξύ των διαδοχικών OFDM συμβόλων, κατά το οποίο καταφθάνουν οι καθυστερημένες εκδοχές του προηγούμενου συμβόλου (λόγω multipath).⁵

Αφετέρου μετατρέπει τη συνέλιξη που πραγματοποιεί το κανάλι στο μεταδιδόμενο σήμα, σε κυκλική συνέλιξη. Αυτή είναι μια πολύ επιθυμητή ιδιότητα αφού καθιστά απλούστερη τη διαδικασία της ισοστάθμισης στο πεδίο της συχνότητας.

Πιο συγκεκριμένα, εάν $x(t)$ είναι το μεταδιδόμενο σήμα στον πομπό και $h(t)$ η κρουστική απόκριση του καναλιού, τότε το λαμβανόμενο σήμα στο δέκτη θα είναι η συνέλιξη του σήματος με την κρουστική απόκριση, δηλαδή

$$y(t) = x(t) * h(t)$$

Στο πεδίο της συχνότητας, η δυαδική της συνέλιξης πράξη είναι ο πολλαπλασιασμός, οπότε για τους μετασχηματισμούς Fourier των παραπάνω σημάτων θα ισχύει

$$Y(f) = X(f) H(f)$$

Αν υποθεθεί τέλεια εκτίμηση καναλιού στο δέκτη, τότε η ισοστάθμιση είναι μία μιγαδική διαίρεση με τη συνάρτηση μεταφοράς του καναλιού, $H(f)$.

Αυτά ισχύουν στα συνεχή απείρου μήκους σήματα. Για να επιτευχθεί το ίδιο αποτέλεσμα στον ψηφιακό κόσμο όπου η διάρκεια των σημάτων είναι πεπερασμένη και γίνεται χρήση του DFT αντί του μετασχηματισμού Fourier, πρέπει να ληφθεί υπόψη ότι στο πεδίο του διακριτού χρόνου, η δυαδική του πολλαπλασιασμού πράξη είναι η κυκλική συνέλιξη. Η τελευταία ορίζεται ως

$$y(n) = x(n) \otimes h(n) = \sum_{p=0}^{N-1} x(p)h((n-p))_N \quad n = 0, 1, \dots, N-1$$

όπου $((x))_N$ συμβολίζει το $x \bmod N$. Η προσθήκη του κυκλικού προθέματος επιτυγχάνει ακριβώς αυτό, μετατρέπει δηλαδή, τη συνέλιξη σε κυκλική συνέλιξη.

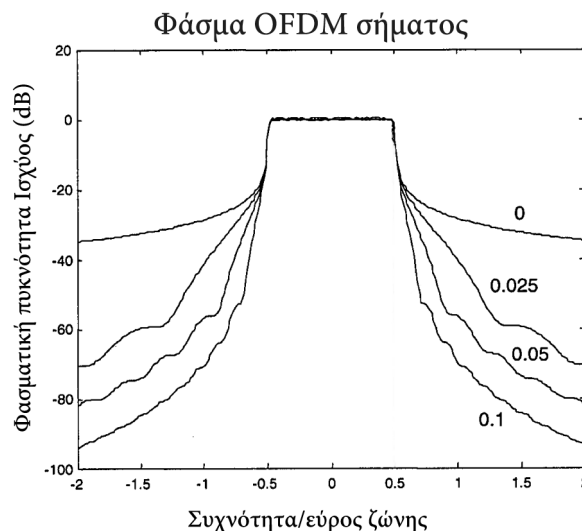
⁴Η ζωνοπερατή διαμόρφωση QAM αποτελείται από κυματομορφές (σύμβολα) της μορφής $u_m(t) = g_T(t)A_{mc}\cos 2\pi f_c t - g_T(t)A_{ms}\sin 2\pi f_c t$, έχοντας επιλέξει ως ορθογώνιες συναρτήσεις βάσεις τις $\psi_1 = \cos 2\pi f_c t$ και $\psi_2 = \sin 2\pi f_c t$. Εναλλακτικά τα σύμβολα μπορούν να αναπαρασταθούν ως $\bar{s}_x = (g_T(t)A_{mc} + jg_T(t)A_{ms})$, όπου τα ζεύγη $(A_{mc} A_{ms})$ επιλέγονται από ένα σύνολο M δυνατών τιμών. Με $g_T(t)$ συμβολίζεται ένας παλμός που καθορίζει τα φασματικά χαρακτηριστικά του τελικού σήματος και στην απλούστερη των περιπτώσεων μπορεί να είναι ο τετραγωνικός παλμός.

⁵Για να επιτευχθεί αυτό θα πρέπει η διάρκεια του CP να είναι μεγαλύτερη από το μέγιστο delay spread του καναλιού.

OFDM Frame

Σε ένα πραγματικό OFDM σύστημα, ένας αριθμός συμβόλων OFDM ομαδοποιείται και αποτελεί ένα frame. Οι διαδικασίες που επιτελούνται σε κάθε frame εισάγουν επιβάρυνση καθώς δεσμεύονται ορισμένα OFDM σύμβολα κατά το οποίο το σύστημα «μαθαίνει» το κανάλι και δεν γίνεται μεταφορά ωφέλιμων δεδομένων. Τα σύμβολα αυτά αποτελούν το προοίμιο (preamble) του frame και ο ρόλος του θα αποσαφηνιστεί παρακάτω, στη συζήτηση για το συγχρονισμό και την εκτίμηση καναλιού.

Μορφοποίηση παλμών Όπως αναφέρθηκε, η τεχνική OFDM χρησιμοποιεί πολύ αποδοτικά το διαθέσιμο φάσμα του καναλιού (ζώνη εκπομπής). Ωστόσο, οι απότομες αλλαγές φάσης που παρουσιάζονται μεταξύ των συμβόλων, οδηγούν σε εξάπλωση του φάσματος σε συχνότητες εκτός της ζώνης μετάδοσης. Αυτό μπορεί να προκαλέσει προβλήματα σε περιβάλλοντα όπου συνυπάρχουν πολλά συστήματα επικοινωνιών, καθώς αυξάνεται ο θόρυβος παρεμβολών από γειτονικά κανάλια. Για την αντιμετώπιση αυτού του φαινομένου χρησιμοποιούνται τεχνικές μορφοποίησης παλμών (pulse shaping), που σα σκοπό έχουν τον περιορισμό του φάσματος στη ζώνη εκπομπής. Στο σχήμα 2.3 παρουσιάζεται ένα τυπικό φάσμα OFDM σήματος, στο οποίο έχει εφαρμοστεί η τεχνική παραθύρου raised cosine, με διάφορες τιμές του συντελεστή roll-off.⁶



Σχήμα 2.3: Φάσμα εκπεμπόμενου OFDM σήματος με raised cosine pulse shaping

2.1.5 Μοντέλο

Με βάση όσα ειπώθηκαν μέχρι τώρα, παρουσιάζεται εδώ το μαθηματικό μοντέλο βασικής ζώνης ενός OFDM συστήματος.

Έστω S_k^λ $k \in 0 \dots N-1$ η ακολουθία των N παράλληλα διαμορφωμένων subsymbol⁷ ομαδοποιημένων ανά N . Ο δείκτης στοιχείου μέσα στην ομάδα k , αναφέρεται στο subcarrier ενός OFDM συμβόλου, ενώ ο δείκτης ομάδας λ , αναφέρεται στο OFDM σύμβολο. Στην παρούσα ανάλυση ο δείκτης λ παραλείπεται, εφόσον παρουσιάζεται το σήμα ενός OFDM συμβόλου. Η ακολουθία των προς μετάδοση subsymbols θα συμβολίζεται εφεξής ως S_k .

⁶Ο συντελεστής roll-off εισέρχεται στον τύπο του raised cosine και προσδιορίζει πόσο απλωμένο θα είναι το φάσμα του παραθύρου σε σχέση με το φάσμα του τετραγωνικού παραθύρου.

⁷Τα subsymbols μπορεί να έχουν διαμορφωθεί από τον ίδιο ή από διαφορετικού τύπου αστερισμό (QAM-4, QAM-16, QAM-64).

Συνεχής χρόνος

Διαμόρφωση Το διαμορφωμένο σήμα ενός OFDM συμβόλου, συνεχούς χρόνου, είναι :

$$x(t) = \sum_{k=0}^{N-1} S_k \exp(j2\pi f_k t) = \sum_{k=0}^{N-1} S_k \phi_k(t) \quad 0 \leq t \leq T_o$$

όπου

$$\phi(t) = \begin{cases} \exp(j2\pi f_k t) & 0 \leq t \leq T_o \\ 0 & \text{αλλού} \end{cases}$$

και $f_k = f_o + \Delta_f$.

Επίσης λόγω της συνθήκης ορθογωνιότητας $T_o \Delta_f = 1$ ισχύει :

$$\frac{1}{T_o} \int_0^{T_o} \phi_k(t) \phi_\lambda^*(t) dx = \delta(k - \lambda) = \begin{cases} 1 & k = \lambda \\ 0 & k \neq \lambda \end{cases}$$

Αποδιαμόρφωση Το αποδιαμορφωμένο σήμα συνεχούς χρόνου, μπορεί να εκφραστεί ως:

$$\begin{aligned} D_k &= \frac{1}{T_o} \int_0^{T_o} x(t) \exp(-j2\pi f_k t) dt = \frac{1}{T_o} \int_0^{T_o} \sum_{\lambda=0}^{N-1} S_\lambda \phi_k(t) \phi_\lambda^*(t) dt = \\ &= \sum_{\lambda=0}^{N-1} S_\lambda \delta(\lambda - k) = S_k \end{aligned}$$

Κανάλι Το σήμα συνεχούς χρόνου, περνώντας από το κανάλι γίνεται

$$y(t) = x(t) * h(t) + n(t)$$

όπου $h(t)$ η κρουστική απόκριση του καναλιού (το οποίο θεωρείται στατικό για τη διάρκεια ενός OFDM συμβόλου), $n(t)$ ο μιγαδικός λευκός προσθετικός θόρυβος Gauss (AWGN).

Διακριτός χρόνος

Στη συνέχεια παρουσιάζεται το μοντέλο διακριτού χρόνου που αφορά στο ψηφιακό σύστημα που υλοποιήθηκε.

Διαμόρφωση Το διαμορφωμένο σήμα διακριτού χρόνου που αντιστοιχεί σε ένα OFDM σύμβολο, με περίοδο δειγματοληψίας T_{sa} είναι :

$$z(n) = x(n T_{sa}) = \sum_{k=0}^{N-1} S_k \exp(j2\pi f_k n T_{sa}) \quad 0 \leq n \leq N - 1$$

Λαμβάνοντας υπόψη ότι $T_{sa} = T_o/N$ και υποθέτοντας σήμα βασική ζώνης, δηλαδή $f_k = 0 \Rightarrow k \Delta_f T_o = k$, προκύπτει ότι

$$z(n) = \sum_{k=0}^{N-1} S_k \exp(j2\pi k \frac{n}{N}) = IDFT\{S_k\}$$

Αποδιαμόρφωση Το αποδιαμορφωμένο σήμα διακριτού χρόνου OFDM, είναι:

$$\begin{aligned} G_k &= \frac{1}{N} \sum_{n=0}^{N-1} z(n) \exp(-j2\pi k \frac{n}{N}) = DFT\{z(n)\} = Z(k) = \\ &= DFT\{IDFT\{S_k\}\} = S_k \end{aligned}$$

Κανάλι Στα πραγματικά συστήματα το διακριτό σήμα $z(n)$ μετατρέπεται σε αναλογικό $x(t)$ μέσω ενός DAC. Στη συνέχεια διαμορφώνεται σε RF σήμα και εκπέμπεται μέσα από το κανάλι. Το λαμβανόμενο σήμα στο δέκτη είναι $y(t)$. Κατόπιν δειγματοληπτείται από έναν ADC και μετατρέπεται σε σήμα διακριτού χρόνου $w(n)$. Εάν υποθεθεί τέλεια δειγματοληψία του λαμβανόμενου σήματος, δηλαδή εάν τα σφάλματα που προκύπτουν από τις διαφορές των συχνοτήτων των ζευγών DAC/ADC θεωρηθούν ως αμελητέα, τότε το λαμβανόμενο σήμα διακριτού χρόνου είναι :

$$w(n) = z(n) \otimes h(n) + n(n)$$

όπου το σήμα $z(n)$, συνελίσσεται κυκλικά με την κρουστική απόκριση διακριτού χρόνου του καναλιού, λόγω της ύπαρξης της κυκλικής επέκτασης, όπως αναφέρθηκε στην §2.1.4.

Συνεπώς, για τους διακριτούς μετασχηματισμούς των παραπάνω σημάτων ισχύει :

$$W(k) = Z(k) H(k) + N(k)$$

όπου $W(k) = DFT\{w(n)\}$, $H(k) = DFT\{h(n)\}$, $N(k) = DFT\{n(n)\}$.

Το αποδιαμορφωμένο σήμα διακριτού χρόνου, μετά το πέρασμά του από το κανάλι, δίνεται από την :

$$\begin{aligned} G_k &= \frac{1}{N} \sum_{n=0}^{N-1} w(n) \exp(-j2\pi k \frac{n}{N}) = DFT\{w(n)\} = W(k) = Z(K) H(k) + N(k) = \\ &= S_k H(K) + N(k) \end{aligned}$$

Δηλαδή, το σήμα αποτελείται από ένα σύνολο παράλληλα μεταδιδόμενων συμβόλων S_k , όπου το καθένα διέρχεται από ένα επίπεδο AWGN κανάλι.

2.1.6 Κανάλι - Θόρυβος

Πηγές θορύβου

Όπως αναφέρθηκε και στο κεφάλαιο 1, η πιο συνηθισμένη μορφή παραμόρφωσης που εισάγει το κανάλι, ενσύρματο ή ασύρματο, είναι ο AWGN. Πρόκειται ένα σύνολο φαινομένων τα οποία μοντελοποιούνται ως μία **στοχαστική διαδικασία** λευκού γκαουσιανού θορύβου με μέση τιμή 0 και μεταβλητότητα N_o^2 . Το μέγεθος που χαρακτηρίζει ποσοτικά το θόρυβο είναι ο σηματοθορυβικός λόγος, $SNR = P/N_o^2$, όπου P η μέση ισχύς του σήματος⁸ και συνήθως εκφράζεται σε dB. Τα φαινόμενα που προκαλούν τον θόρυβο προέρχονται τόσο από πηγές που βρίσκονται μέσα στο κανάλι,⁹ όσο και από παράγοντες θορύβου του ίδιου του συστήματος. Κατ' επέκταση ο θόρυβος διακρίνεται σε εξωτερικό, που περιλαμβάνει τις πηγές θορύβου μέχρι τον DAC και σε εσωτερικό, που περιλαμβάνει τις πηγές θορύβου μέχρι και το υποσύστημα αποδιαμόρφωσης κατά QAM (symbol demapper).¹⁰ Προφανώς, το μέγεθος του θορύβου στην είσοδο του demapper, είναι το άθροισμα του μεγέθους όλων των πηγών θορύβου, εξωτερικών και εσωτερικών.

⁸ Η μέση ισχύς του θορύβου είναι N_o

⁹ Μία σημαντική πηγή θορύβου σε περιβάλλοντα που συνυπάρχουν πολλά τηλεπικοινωνιακά συστήματα, είναι ο θόρυβος από γειτονικές παρεμβολές. Δηλαδή από συστήματα που εκπέμπουν στην ίδια RF συχνότητα λειτουργίας του δέκτη, με αποτέλεσμα να προστίθενται στο σήμα προς αποδιαμόρφωση. Εν τέλει συνυπολογίζονται στο μοντέλο AWGN.

¹⁰ Στον εξωτερικό θόρυβο συγκαταλέγονται και οι αναλογικές πηγές θορύβου του συστήματος, όπως η RF ενισχυτική βαθμίδα του δέκτη.

Εξωτερικός θόρυβος

Η κύρια επίπτωση του εξωτερικού θορύβου AWGN στις ψηφιακές διαμορφώσεις, όπως το OFDM, είναι ο διασκορπισμός των σημείων του αστερισμού στον δέκτη, γεγονός που εισάγει σφάλματα κατά τη διαδικασία της αποδιαμόρφωσης κατά QAM. Το φαινόμενο γίνεται έκδηλο με την παρατήρηση ενός διαγράμματος διασποράς (scatterplot)¹¹ των ληφθέντων συμβόλων στην είσοδο του demapper. Τα σύμβολα διασπείρονται υπό μορφή νέφους, γύρω από το εκάστοτε σημείο του αστερισμού που αντιστοιχεί στο καθένα από αυτά. Όσο μικραίνει το SNR , τόσο απλώνει το «νέφος» των συμβόλων. Το φαινόμενο αυτό, εισάγει ασάφεια στην αποδιαμόρφωση των συμβόλων. Κατ' επέκταση, μπορεί να ληφθεί λανθασμένη απόφαση για ένα σύμβολο, εάν αυτό βρεθεί σε περιοχή ενός γειτονικού σημείου του αστερισμού. Ο ρυθμός σφαλμάτων συμβόλων (SEr) είναι ένα μέτρο της πιθανότητας λανθασμένης απόφασης για ένα σύμβολο. Το SEr γενικά αυξάνεται, όσο μεγαλώνει η διασπορά των συμβόλων, δηλαδή όσο μικραίνει το SNR .

Εσωτερικός θόρυβος

Εσωτερικό SNR Εν γένει, το scatterplot των συμβόλων που εισέρχονται στον demapper, απεικονίζει ένα άθροισμα πηγών θορύβου, τόσο εξωτερικών, όσο και εσωτερικών. Συνεπώς, αν υπολογιστεί ο σηματοθορυβικός λόγος στην είσοδο του demapper, θα βρεθεί μικρότερος από το SNR του καναλιού. Αυτός ο λόγος θα καλείται εφεξής «εσωτερικό SNR » ή $ISNR$. Για την αποτίμηση της επίδοσης του συστήματος, είναι προτιμότερο το μέγεθος του «εσωτερικού θορύβου», $INNOISE = SNR - ISNR$.¹² Το $ISNR$ εκφραζόμενο σε dB, μπορεί να υπολογιστεί ως :

$$ISNR = 10 \log_{10} \left(\frac{P_{const}}{P_{err}} \right)$$

όπου

$$\begin{aligned} S_k \quad k \in 0 \dots M-1 \quad & \text{τα σύμβολα του αστερισμού της διαμόρφωσης QAM-M,} \\ P_{const} = E_k[|S_k|^2] \quad & \text{η μέση ισχύς του αστερισμού,} \\ R_m^{(k)} \quad & \text{η ακολουθία των ληφθέντων συμβόλων με } R_m^{(k)} = S_k, \\ P_{err} = E_m[|R_m^{(k)} - S_k|^2] \quad & \text{η μέση ισχύς του σφάλματος} \end{aligned}$$

Επιπτώσεις θορύβου στην εκτίμηση καναλιού Μία σημαντική πηγή εσωτερικού θορύβου είναι η διαδικασία της εκτίμησης καναλιού. Η τελευταία, μπορεί να επηρεαστεί αρνητικά από την παρουσία (εξωτερικού) θορύβου, καθώς εισάγεται ασάφεια στους εκτιμώμενους συντελεστές που περιγράφουν το κανάλι. Μάλιστα, όσο μικρότερο το SNR , τόσο μεγαλύτερη η ασάφεια των συντελεστών.¹³ Η ισοστάθμιση των συμβόλων γίνεται με λανθασμένους συντελεστές, με αποτέλεσμα τον περαιτέρω διασκορπισμό τους σε «νέφη», γύρω από τα σημεία του αστερισμού. Το φαινόμενο αυτό, είναι εμφανές στο scatterplot των συμβόλων μετά την ισοστάθμιση καναλιού.

Σφάλματα συγχρονισμού Ένας άλλος παράγοντας εσωτερικού θορύβου, προέρχεται από τα σφάλματα συγχρονισμού στο χρόνο και στη συχνότητα. Τα σφάλματα συγχρονισμού στο χρόνο, εισάγουν ασάφεια ως προς το σημείο έναρξης του DFT η οποία, εάν ξεπεράσει κάποια όρια, εισάγει στο αποδιαμορφωμένο σήμα παρεμβολές ISI και ICI. Το αποτέλεσμα της παρουσίας ISI είναι η αποδιαμόρφωση samples από το επόμενο σύμβολο, ενώ του ICI, είναι η μερική απώλεια της ορθογωνιότητας των subcarrier. Στα σφάλματα συγχρονισμού στη συχνότητα, το φάσμα του σήματος ολισθαίνει, με αποτέλεσμα τη μερική

¹¹Scatterplot : Ένα διάγραμμα στο μιγαδικό επίπεδο στο οποίο απεικονίζεται κάθε ληφθέν σύμβολο. Από την πυκνότητα των συμβόλων πάνω στο διάγραμμα, αντιλαμβάνεται κανείς τη συχνότητα εμφάνισής τους γύρω από συγκεκριμένες περιοχές.

¹²Το $INNOISE$ μπορεί να αναφέρεται και ως επιδείνωση του SNR (SNR degradation)

¹³Για την εκτίμηση καναλιού υπάρχουν πολλοί αλγόριθμοι που διαφέρουν σε απόδοση και πολυπλοκότητα και παρουσιάζουν διαφορετική ανοχή στην παρουσία θορύβου.

απώλεια της ορθογωνιότητας των subcarrier και την εισαγωγή ICI. Και τα δύο είδη παρεμβολών έχουν μεγάλες επιπτώσεις στην ορθή λήψη OFDM σήματος, αλλά η μεγαλύτερη ευαισθησία παρουσιάζεται στα σφάλματα συχνότητας, γιατί εισάγεται παρεμβολή σε όλα τα subcarrier. Εν τέλει, τα σφάλματα συγχρονισμού, συνεισφέρουν στην εξάπλωση του νέφους του αστερισμού και στην αύξηση του εσωτερικού θορύβου. Εδώ, αξίζει να σημειωθεί πως το SNR του καναλιού, παίζει καθοριστικό ρόλο στην επίδοση των αλγορίθμων συγχρονισμού, οπότε επηρεάζει έμμεσα τον εσωτερικό θόρυβο που δημιουργείται από τα σφάλματα συγχρονισμού.

Ψηφιακός θόρυβος

Στα ψηφιακά συστήματα εμφανίζονται δύο μεγάλες κατηγορίες θορύβου, οι οποίες είναι εγγενείς στη φύση των συστημάτων αυτών. Η πρώτη κατηγορία σχετίζεται με τα σφάλματα κβαντισμού των ψηφιακών μεγεθών που έχουν προέλθει μετά από την ψηφιοποίηση αναλογικών δεδομένων, ενώ η δεύτερη είναι σχετική με την πεπερασμένη ακρίβεια αναπαράστασης των αριθμών σε έναν ψηφιακό υπολογιστή κατά την τέλεση μαθηματικών πράξεων.

Σφάλματα κβαντισμού Προέρχονται από την περιορισμένη διακριτική ικανότητα των DAC/ADC. Κάποιες στάθμες πραγματικών σημάτων, δεν μπορούν να αναπαρασταθούν επακριβώς και στρογγυλοποιούνται στην πιο κοντινή τιμή με αποτέλεσμα την εισαγωγή ψηφιακού θορύβου στο σήμα. Το SNR λόγω ψηφιακού θορύβου (Digital-Noise -SNR - $DSNR$) για το σήμα A , που προκαλείται από το σφάλμα κβαντισμού Q , είναι :

$$DSNR = 20 \log_{10} \left(\frac{A_{rms}}{Q_{rms}} \right)$$

Σημειώνεται ότι το Q εξαρτάται τόσο από το είδος κβαντισμού, όσο και από το είδος του σήματος A . Εάν υποτεθεί ημιτονοειδές σήμα εισόδου για ένα σύστημα DAC με fixed point ανάλυση της τάξης των N bit και κβαντισμό τύπου «σκάλας», τότε $A_{rms} = \frac{2^N}{2\sqrt{2}}$ και $Q_{rms} = \frac{1}{2\sqrt{3}}$, εφόσον τώρα το Q είναι μία πριονωτή συνάρτηση. Άρα για το $DSNR$ προκύπτει :

$$DSNR = 20 \log_{10} \left(2^N \frac{\sqrt{3}}{\sqrt{2}} \right) \approx 6.02 + 1.761 N$$

Σφάλματα ακρίβειας Προέρχονται από την πεπερασμένη ακρίβεια του συστήματος αριθμητικής που χρησιμοποιείται (π.χ. fixed point) για την τέλεση των διαφόρων πράξεων κατά την αποδιαμόρφωση του σήματος. Η επιλογή της σωστής ακρίβειας για κάθε πράξη, έτσι ώστε να περιορίζεται το σφάλμα και ταυτόχρονα να μη γίνεται σπατάλη υλικού, θα μπορούσε να χαρακτηριστεί ως τέχνη. Ο κλάδος της επιστήμης υπολογιστών που ασχολείται με την αριθμητική υπολογιστών, έχει να προσφέρει μία πληθώρα εργαλείων ανάλυσης σφάλματος και μεθόδων για την αποδοτική υλοποίηση βασικών πράξεων και συναρτήσεων σε έναν ψηφιακό υπολογιστή.

Σχόλια για τις πηγές θορύβου

Ανακεφαλαιώνοντας, η συνολική ισχύς του θορύβου στην είσοδο του αποδιαμορφωτή κατά QAM στο δέκτη ενός τηλεπικοινωνιακού συστήματος, είναι αποτέλεσμα των διαφόρων ειδών θορύβου που εισάγονται στη διαδρομή του σήματος. Το παρατηρήσιμο αποτέλεσμα είναι η μείωση του $ISNR$ σε σχέση με το SNR του σήματος μετά τη διέλευσή του από το κανάλι.

Σε ένα τηλεπικοινωνιακό σύστημα υπάρχουν, πάντα, τρεις βασικές πηγές εξωτερικού θορύβου: Το κανάλι, οι αναλογικές βαθμίδες στον πομπό και στο δέκτη και η διαδικασία του κβαντισμού. Ο θόρυβος του καναλιού, δεν είναι δυνατό να περιοριστεί από το σύστημα καθώς εξαρτάται από το περιβάλλον μετάδοσης. Υπάρχουν ωστόσο, αναλυτικά μοντέλα για μία πληθώρα κατηγοριών καναλιών, τα οποία μπορούν να

προσδιορίσουν τα βασικά στατιστικά χαρακτηριστικά του θορύβου, ώστε αυτά να ληφθούν υπόψη κατά τη διαδικασία της σχεδίασης.

Οι άλλες δύο πηγές θορύβου εξαρτώνται από παραμέτρους που αφορούν το σύστημα και η τροποποίησή τους είναι δύσκολη καθώς περιλαμβάνει την αλλαγή της αρχιτεκτονικής του. Για τη βελτίωση του αναλογικού θορύβου μπορεί να γίνει χρήση καλύτερων, από άποψη ποιότητας, αναλογικών ηλεκτρονικών συστημάτων στην RF βαθμίδα στο δέκτη. Ενώ για τη μείωση του ψηφιακού θορύβου που προέρχεται από τον κβαντισμό, μπορεί να γίνει χρήση ADC μεγαλύτερης διακριτικής ανάλυσης.

Αν έχει προσδιοριστεί η κατηγορία καναλιών για την οποία προορίζεται να δουλέψει το σύστημα και αν έχει επιλεγεί σύστημα ADC και αναλογικών βαθμίδων, τότε όλες οι πηγές εξωτερικού θορύβου μπορούν να θεωρηθούν γνωστές. Συνεπώς μπορεί να προσδιοριστεί η μέση ισχύς του θορύβου ως το άθροισμα της μέσης ισχύος όλων των πηγών εξωτερικού θορύβου. Αυτή η τιμή ορίζεται ως το *κάτω όριο θορύβου (noise floor)*, όσων αφορά το ψηφιακό σύστημα. Δηλαδή η ισχύς του θορύβου δεν δύναται να πέσει κάτω από αυτό το όριο.

Αντιθέτως, ο εσωτερικός θόρυβος, που προέρχεται από τις λειτουργίες που επιτελούνται για την αποδιαμόρφωση του σήματος, μπορεί να μειωθεί με αποφάσεις που λαμβάνονται κατά τη σχεδίαση του ψηφιακού συστήματος. Για παράδειγμα, ο θόρυβος συγχρονισμού και ο θόρυβος σφάλματος εκτίμησης καναλιού μπορούν να βελτιωθούν με χρήση αποδοτικότερων αλγορίθμων, πληρώνοντας όμως το τίμημα σε αυξημένη πολυπλοκότητα.

Πάντως, η συγκεκριμενοποίηση της στάθμης θορύβου ανά υποσύστημα, είναι επίπονη διεργασία και προϋποθέτει ανάλυση εις βάθος των χαρακτηριστικών του συστήματος. Επιχειρώντας μία σύγκριση, οι πηγές θορύβου σε ένα τυπικό σύστημα θα μπορούσαν να καταταγούν από την μικρότερη στη μεγαλύτερη ισχύ ως εξής:

Θόρυβος κβαντισμού < Σφάλματα ακρίβειας < Σφάλμα εκτίμησης καναλιού < Σφάλμα συγχρονισμού < Εξωτερικός θόρυβος.

Multipath fading

Όπως αναφέρθηκε και στο πρώτο κεφάλαιο μία από τις βασικές παραμορφώσεις που εισάγει το ασύρματο κανάλι (ραδιοδιάυλος) στα μικροκυματικά συστήματα είναι η πολυδιαδρομική διάδοση (multipath propagation). Προκύπτει όταν στην κεραία λήψης του δέκτη, μαζί με το εξασθενημένο σήμα του πομπού φτάνουν και πολλαπλά καθυστερημένα αντίγραφα αυτού, από διάφορες κατευθύνσεις και μάλιστα με διαφορετικά πλάτη, φάσεις, συχνότητες και καθυστερήσεις. Αυτά προστίθενται διανυσματικά και προκαλούν φαινόμενα διαλείψεων (fading), δηλαδή μεγάλων κυματώσεων στην περιβάλλουσα (και την ισχύ) του λαμβανόμενου σήματος, της τάξης των 40dB για χωρικές αποστάσεις συγκρίσιμες με το μήκος κύματος. Επιπλέον όταν υπάρχει σχετική κίνηση του δέκτη με τον πομπό το κανάλι αλλάζει με το πέρασμα του χρόνου (χρονομεταβλητό). Ουσιαστικά το multipath εισάγει διασπορά του σήματος τόσο στο χρόνο όσο και στη συχνότητα.

Τα βασικά μεγέθη που χαρακτηρίζουν το προφίλ multipath ενός καναλιού και απαντώνται μετά από μετρήσεις είναι:

1. Διασπορά καθυστέρησης (delay spread) $[\tau_d]$ - όταν μεταδίδεται ένας κρουστικός παλμός μέσα από το κανάλι, στο δέκτη καταφθάνουν διαφορετικές εξασθενημένες και καθυστερημένες εκδοχές του με αποτέλεσμα το τελικό σήμα να είναι ένας παλμός με διάρκεια τ_d .
2. Εύρος ζώνης συνοχής (coherence bandwidth) $[B_c]$ - αντίστοιχο στο πεδίο της συχνότητας μέγεθος για το delay spread. Είναι το εύρος ζώνης κατά το οποίο οι φασματικές συνιστώσες έχουν παρόμοιο κέρδος και γραμμική φάση. Είναι ένα στατιστικό μέτρο του εύρους της περιοχής όπου ο διάυλος θεωρείται επίπεδος (flat). Ένας απλός, εμπειρικός τύπος υπολογισμού του είναι ο $B_c = 1/\{\tau_d\}_{RMS}$.
3. Εξάπλωση Doppler (Doppler spread) $[B_D]$ - λόγω της σχετικής κίνησης του δέκτη ως προς τον πομπό ή αντικειμένων του περιβάλλοντος ως προς το δέκτη, παρατηρείται ολίσθηση Doppler στο φά-

σμα του ληφθέντος σήματος, η οποία εξαρτάται αναλογικά από την ταχύτητα της συνισταμένης των κινήσεων. Το αποτέλεσμα είναι η διασπορά του φάσματος.

4. Χρόνος συνοχής (coherence time) $[C_T]$ - αντίστοιχο στο πεδίο του χρόνου μέγεθος για το Doppler spread. Αποτελεί στατιστικό μέτρο του χρονικού διαστήματος για το οποίο η κρουστική απόκριση παραμένει σταθερή. Προσδιορίζεται εμπειρικά ως $C_T = 1/B_D$.

Οι πιθανοί χαρακτηρισμοί ενός multipath καναλιού σε σχέση με το είδος διαλείψεων που προκαλεί σε ένα σήμα με χρόνο συμβόλου T_O και εύρος ζώνης W φαίνεται στον Πίνακα 2.1.

Χαρακτηρισμός	Συνθήκη
Επίπεδο (flat)	$W < B_c$ ή $T_S > \tau_d$
Συχνοεπιλεκτικό (frequency selective)	$W > B_c$ ή $T_S < \tau_d$
Βραδέως μεταβαλλόμενο (slow changing)	$W < B_D$ ή $T_S > C_T$
Ταχέως μεταβαλλόμενο (fast changing)	$W > B_D$ ή $T_S < C_T$

Πίνακας 2.1: Χαρακτηρισμοί multipath καναλιών

2.1.7 Παράμετροι OFDM

Εδώ αναφέρονται τα κυριότερα μεγέθη που χαρακτηρίζουν ένα OFDM σύστημα. Η επιλογή των τιμών τους, συνίσταται στη διαδικασία της σχεδίασης του τηλεπικοινωνιακού συστήματος.

Εύρος ζώνης W Το εύρος του φάσματος που καταλαμβάνει το τελικό σήμα στο κανάλι.

Διάρκεια ωφέλιμου συμβόλου (useful symbol duration) T_o Η χρονική διάρκεια του OFDM συμβόλου χωρίς την κυκλική επέκταση. Η παράμετρος αυτή επιλέγεται έτσι, ώστε να είναι μεγαλύτερη από το delay spread και μικρότερη από το coherence time του καναλιού.

Μήκος ωφέλιμου συμβόλου N_o Το μήκος του OFDM συμβόλου σε samples. Από τον τύπο του DFT και δεδομένου ότι δεν χρησιμοποιείται υπερ-δειγματοληψία (oversampling) στην έξοδο του DFT, προκύπτει ότι $N_o = N$. Επιπλέον ισχύει $T_o = N_o T_{sa}$

Διάρκεια καναλιού T_h Το μέγιστο delay spread του καναλιού, δηλαδή η μέγιστη διάρκεια που θεωρητικά μπορεί να έχει η κρουστική απόκριση του καναλιού $h(t)$.

Μήκος καναλιού N_h Το μήκος του καναλιού σε samples. Ισχύει $N_h = T_h/T_{sa}$.

Διάρκεια κυκλικής επέκτασης T_{CP} Η χρονική διάρκεια της κυκλικής επέκτασης. Επιλέγεται έτσι ώστε να είναι μεγαλύτερη από τη διάρκεια του καναλιού, $T_{CP} > T_h$.

Μήκος κυκλικής επέκτασης N_{CP} Το μήκος του κυκλικού προθέματος σε samples. Ισχύει $T_{CP} = N_{cp} T_{sa}$.

Διάρκεια συμβόλου (symbol duration) T_s Η χρονική διάρκεια ενός OFDM συμβόλου. Ισχύει $T_s = T_o + T_{CP}$

Απόσταση μεταξύ των subcarrier (subcarrier spacing) Δ_f Η απόσταση στο πεδίο της συχνότητας δύο διαδοχικών subcarrier. Όπως αναφέρθηκε η απόσταση μεταξύ των subcarriers είναι $\Delta_f = 1/T_o$, προκειμένου να ισχύει η συνθήκη ορθογωνιότητας.

Πλήθος subcarrier N Ο αριθμός των διαδοχικών υπο-φερόντων (subcarriers). Το συνολικό εύρος ζώνης είναι $W = N\Delta_f$. Επίσης είναι το μέγεθος του FFT/IFFT που χρησιμοποιείται.

Μέγεθος ψηφιακής διαμόρφωσης M Το μέγεθος του αστερισμού της ψηφιακής διαμόρφωσης χρησιμοποιείται (M-αδική διαμόρφωση QAM-4, QAM-16, QAM-64). Η επιλογή του βασίζεται στην όσο το δυνατόν αποδοτικότερη εκμετάλλευση του φάσματος και της ισχύος. Όταν το κανάλι παρουσιάζει υψηλό SNR χρησιμοποιούνται αστερισμοί υψηλής φασματικής απόδοσης (QAM-64), ενώ στην περίπτωση που το SNR είναι χαμηλό χρησιμοποιούνται απλούστεροι αστερισμοί (QPSK).

Περίοδος δειγματοληψίας T_{sa} Η περίοδος δειγματοληψίας του DAC/ADC. Με δεδομένες τις επιλογές τιμών για το πλήθος των subcarrier, N και της χρονικής διάρκειας του ωφέλιμου OFDM συμβόλου T_o και για απλή δειγματοληψία, ισχύει: $T_{sa} = T_o/N$ και $\Delta_f = W$.

Ρυθμός δειγματοληψίας F_{sa} Η συχνότητα δειγματοληψίας του DAC και ιδανικά του ADC. Ισχύει $F_{sa} = 1/T_{sa}$.

Μήκος frame R Η επιλογή του μήκους του frame εξαρτάται από το πόσο γρήγορα μεταβάλλεται το κανάλι (coherence time). Ένα ταχέως μεταβαλλόμενο κανάλι επιβάλλει μικρό μήκος frame και το αντίστροφο.

2.2 Μέρος Β - Πρακτική Υλοποίηση Συστήματος

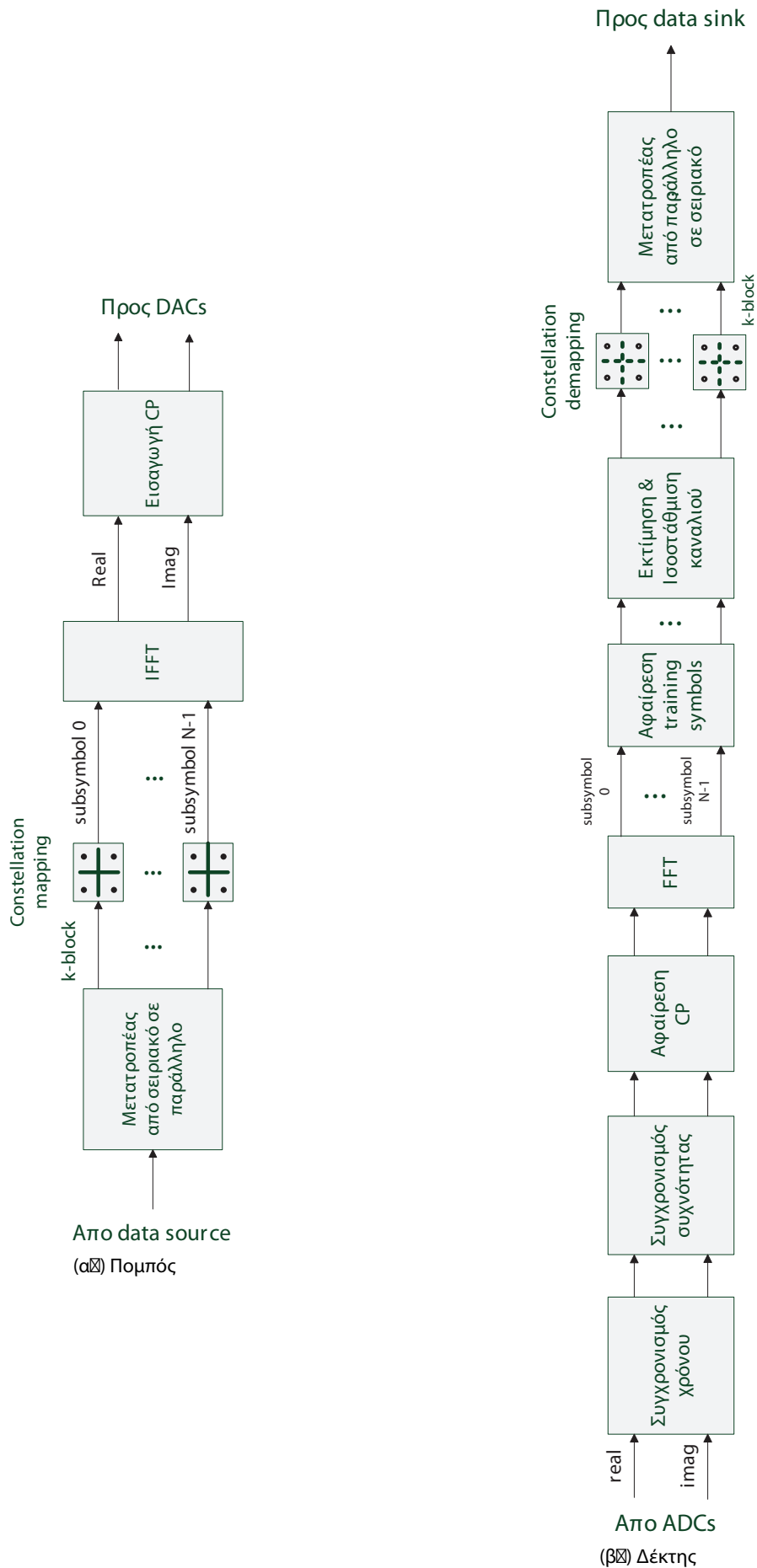
Η συζήτηση, από το σημείο αυτό και εξής, θα συνεχιστεί πάνω σε θέματα που αφορούν στο ψηφιακό σύστημα που υλοποιήθηκε. Στο σχήμα 2.4 παρουσιάζεται το **λειτουργικό διάγραμμα** του συστήματος.¹⁴

2.2.1 Παραδοχές

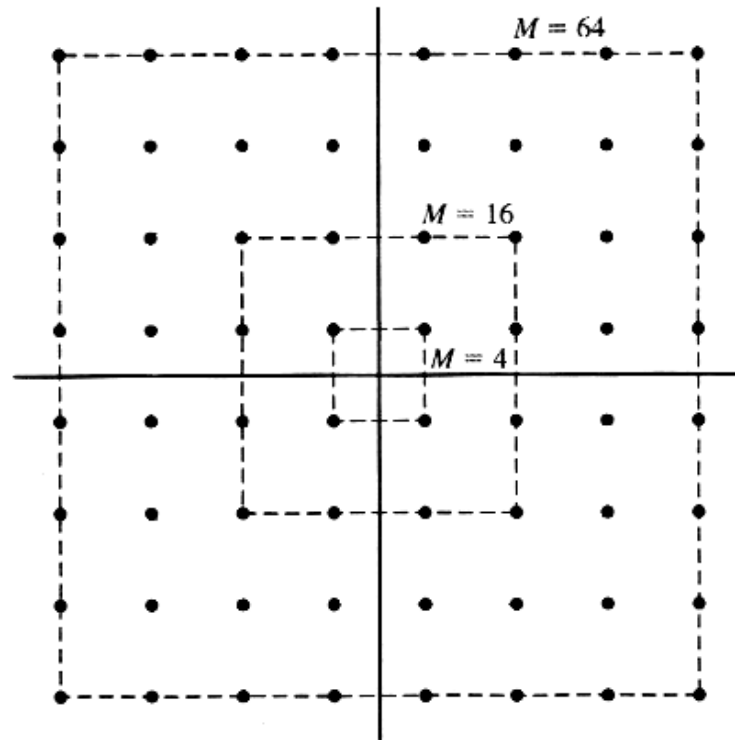
Slow fading, multipath κανάλι Ως προδιαγραφή για το περιβάλλον λειτουργίας, επιλέγεται slow fading frequency selective multipath κανάλι. Δηλαδή, το κανάλι είναι συχνοεπιλεκτικό, βραδέως μεταβαλλόμενο. Συγκεκριμένα, θεωρείται στατικό για τη διάρκεια ενός frame, δηλαδή δεν υπάρχουν μετρήσιμα φαινόμενα Doppler. Ακόμα, θεωρείται ότι η συνολική ολίσθηση συχνότητας -λόγω διαφοράς στις συχνότητες των RF ταλαντωτών- είναι σχεδόν σταθερή και μικρή. Για την ακρίβεια θεωρείται ότι είναι $|\Delta_f^s| < \Delta_f$, δηλαδή η ολίσθηση συχνότητας είναι μικρότερη από την απόσταση μεταξύ δύο subcarrier.

Τετραγωνικό σχήμα αστερισμών Λόγω του ιδιαίτερου πλεονεκτήματος που παρουσιάζουν στη διαδικασία της αποδιαμόρφωσης, χρησιμοποιούνται τετραγωνικοί αστερισμοί QAM. Δηλαδή τα σύμβολα είναι διατεταγμένα σε έναν κάρναβο πάνω στο μιγαδικό επίπεδο (Σχ 2.5).

¹⁴Για μία απεικόνιση των βαθμίδων που απαρτίζουν το σύστημα, ο αναγνώστης παραπέμπεται στο κεφάλαιο 3.

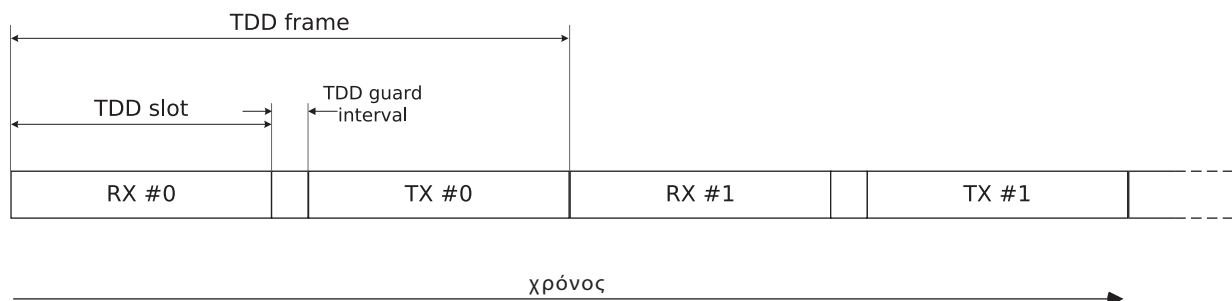


Σχήμα 2.4: Λειτουργικό διάγραμμα του υλοποιηθέντος συστήματος OFDM



Σχήμα 2.5: Τετραγωνικοί αστερισμοί QAM

Τρόπος μετάδοσης - Πολυπλεξία Ως βασική προδιαγραφή για τον τρόπο μετάδοσης, θεωρείται το simplex mode. Εδώ, το ίδιο κύκλωμα μπορεί να ρυθμιστεί να λειτουργεί σταθερά είτε ως πομπός είτε ως δέκτης. Η επιλογή του τρόπου μετάδοσης είναι σταθερή σε κάθε «στιγμιότυπο»¹⁵ του κυκλώματος. Ωστόσο, καθώς το σύστημα αποτελείται από ένα ενιαίο κύκλωμα πομποδέκτη, θα μπορούσε να λειτουργήσει σε λειτουργία «πολυπλεξίας διαίρεσης χρόνου» (TDD). Σε αυτόν τον τρόπο λειτουργίας χρησιμοποιείται ένα κανάλι διπλής κατεύθυνσης, τόσο για τη λήψη όσο και για την εκπομπή, οι οποίες πραγματοποιούνται σε μη επικαλυπτόμενα χρονικά παράθυρα TX και RX. Ένα ζεύγος από δύο διαδοχικά παράθυρα -TX και RX- με την παρεμβολή ανάμεσά τους ενός κενού διαστήματος, ορίζει ένα TDD frame (Σχ. 2.6). Ωστόσο, δεν έχει υλοποιηθεί πλήρως ο ελεγκτής πολυπλεξίας (TDD controller), καθώς δεν θεωρείται στις προδιαγραφές της παρούσας σχεδίασης.



Σχήμα 2.6: TDD frame

¹⁵ Ο όρος «στιγμιότυπο» ενός κυκλώματος διακρίνει τη σχεδίαση ενός κυκλώματος, από την υλοποίησή του. Θα εξηγηθεί καλύτερα στο Κεφάλαιο 3

Σχήμα παλμών Οι παλμοί είναι τετραγωνικής μορφής καθώς δεν εφαρμόζεται κάποια τεχνική μορφοποίησης παλμών (pulse shaping).

Μη υλοποιημένες λειτουργίες Υλοποιείται το Uncoded OFDM, δηλαδή δεν περιλαμβάνονται τα στάδια της κωδικοποίησης και αποκωδικοποίησης καναλιού. Ως πηγή δεδομένων θεωρείται μία ακολουθία τυχαίων ψηφιακών δεδομένων. Επίσης, δεν λαμβάνεται μέριμνα για τον περιορισμό του PAPR.¹⁶

2.2.2 Παράμετροι

Οι παράμετροι OFDM του συστήματος που υλοποιήθηκε, συνοψίζονται στον Πίνακα 2.2.¹⁷ Αξίζει να σημειωθεί, πως το κύκλωμα του πομποδέκτη έχει υλοποιηθεί με τέτοιο τρόπο ώστε να είναι εύκολη η αλλαγή τους, καθιστώντας απλή την παραμετροποίηση και την αλλαγή των χαρακτηριστικών του συστήματος.

Εύρος ζώνης	W	20 MHz
Αριθμός subcarrier/ μέγεθος FFT	N	128
Απόσταση subcarrier	Δ_f	156,25 KHz
Διάρκεια ωφέλιμου συμβόλου	T_o	6,4 μ s
Διάρκεια συμβόλου OFDM	T_s	8 μ s
Μήκος ωφέλιμου συμβόλου	N_o	128 samples
Μήκος CP	N_{CP}	32 samples
Μήκος καναλιού	T_h	16 samples
Μήκος frame	R	10 symbols

Πίνακας 2.2: Παράμετροι OFDM του συστήματος που υλοποιήθηκε

Διάνυσμα διαμόρφωσης QAM \vec{M} Στο σύστημα, υπάρχει δυνατότητα υποστήριξης διαφορετικής διαμόρφωσης ανά subcarrier, όπως προσδιορίζεται από ένα διάνυσμα \vec{M} , μήκους N . Οι πιθανές τιμές του κάθε στοιχείου του είναι 2,4,6 για τους τύπους διαμόρφωσης QAM-4, QAM-16, QAM-64, αντίστοιχα. Πρέπει να σημειωθεί ότι το \vec{M} επιλέγεται κατά τη σχεδίαση σύμφωνα με τις προδιαγραφές του καναλιού, δηλαδή είναι σταθερό για ένα instance του κυκλώματος.

Στην τρέχουσα σχεδίαση είναι:

$$\vec{M} = [4, 4, \dots, 4]$$

Ωστόσο, σε ένα πρακτικό σύστημα OFDM, που υλοποιείται ένας προσαρμοστικός αλγόριθμος επιλογής του τύπου διαμόρφωσης ανά subcarrier βάσει της κατάστασης του καναλιού, το \vec{M} θα υπολογίζεται κατά τη λειτουργία του συστήματος.

Θέματα υποσυστημάτων

2.2.3 Συγχρονισμός στο Χρόνο

Ένα από τα πιο θεμελιώδη προβλήματα που πρέπει να επιλυθούν σε ένα πρακτικό OFDM σύστημα, είναι ο συγχρονισμός στο χρόνο και στη συχνότητα (εκτίμηση ολίσθησης συχνότητας, για την οποία θα γίνει αναφορά παρακάτω). Ο **συγχρονισμός στο χρόνο** (timing synchronization), ή απλά συγχρονισμός,

¹⁶Αναφορικά, κάποιοι τρόποι μείωσης του PAPR είναι: η χρήση προ-κωδικοποίησης ώστε να αυξηθεί τεχνητά η «τυχασιότητα» των δεδομένων, η βελτιστοποίηση της μεταδιδόμενης ισχύος ανά subcarrier ώστε να αποφευχθεί η ενισχυτική συμβολή τους και η εισαγωγή γνωστής παραμόρφωσης στο σήμα ώστε να αντισταθμιστεί η παραμόρφωση που εισάγει ο πομπός.

¹⁷Οι παράμετροι που αφορούν το συγχρονισμό (M_{rthres} , M_{span} , T_h) θα οριστούν στην §2.2.3.

αφορά την ανίχνευση της αρχής κάθε λαμβανόμενου OFDM συμβόλου. Συνήθως¹⁸ αυτό πραγματοποιείται με την εισαγωγή κάποιων «εκπαιδευτικού» τύπου συμβόλων (training symbols) στην αρχή του κάθε frame.¹⁹ Τα training symbols έχουν cyclic prefix όπως και τα σύμβολα δεδομένων (data symbols), πλην όμως δεν φέρουν ωφέλιμο φορτίο πληροφορίας (payload). Το μέγεθος του frame πρέπει να επιλεγεί με γνώμονα την ελαχιστοποίηση της επιβάρυνσης, με ταυτόχρονη προστασία από σφάλματα συγχρονισμού και ολίσθησης συχνότητας.

Ο στόχος της διαδικασίας του συγχρονισμού είναι η εύρεση του σωστού σημείου, στο χρόνο, για την έναρξη του DFT. Λανθασμένη επιλογή αυτού του σημείου οδηγεί σε ICI ή ISI, αφού χάνεται τμηματικά η ορθογωνιότητα των subcarrier. Το ISI συμβαίνει όταν τμήμα του επόμενου συμβόλου θεωρείται ως τρέχων, ενώ το ICI συμβαίνει όταν υπάρχει απώλεια samples από το τρέχων σύμβολο.

Αν υποθεθεί ότι:

το μήκος του καναλιού είναι N_h , με $N_h < N_{CP}$,
η αρχή ενός OFDM συμβόλου στο χρόνο είναι θ ,
η εκτίμηση της αρχής είναι $\hat{\theta}$,
το σημείο έναρξης του DFT είναι $\hat{\theta} + M$,

τότε η ασφαλής περιοχή για την εκτίμηση της αρχής του συμβόλου ώστε να αποφευχθούν τα ISI και ICI είναι

$$\theta + N_h < \hat{\theta} + M < \theta + N_{CP}$$

Αν το $\hat{\theta}$ βρίσκεται σε αυτή την περιοχή, η μόνη παρενέργεια του σφάλματος εκτίμησης είναι μια στροφή φάσης, η οποία μπορεί εύκολα να αντισταθμιστεί από το υποσύστημα εκτίμησης καναλιού (channel estimator). Όσο μεγαλύτερο το μήκος του cyclic prefix, τόσο μεγαλύτερο και το περιθώριο ανοχής σε σφάλματα συγχρονισμού (Σχήμα 2.7).²⁰

Ως βέλτιστο σημείο συγχρονισμού επιλέγεται το μέσον της περιοχής ασφαλείας, δηλαδή

$$(\hat{\theta} + M)_{opt} = \theta + \frac{N_{CP} - N_h}{2}$$

Με αυτή την επιλογή γίνεται σωστή εκμετάλλευση της ανοχής του OFDM σε σφάλματα συγχρονισμού (ελαχιστοποίηση σφαλμάτων ICI και ISI).

Αλγόριθμος

Ο αλγόριθμος που επιλέχτηκε για την παρούσα υλοποίηση είναι των Schmidl και Cox (S&C) όπως περιγράφεται στο [SC97] με μία τροποποίηση που εισήγαγαν αργότερα οι Bargava, Minn και Zeng στο [MZB00]. Πρόκειται για έναν εύρωστο (robust) αλγόριθμο συγχρονισμού, τόσο στο χρόνο, όσο και στη συχνότητα. Γεγονός που εξασφαλίζει άριστη απόδοση σε βεβαρυμένα, από τηλεπικοινωνιακής απόψεως, περιβάλλοντα (έντονη παρουσία AWGN, φαινόμενα multipath κλπ.).

Για την εύρεση του σωστού σημείου συγχρονισμού, εισάγεται στην αρχή του κάθε frame ένα training symbol. Αποτελείται από δύο ίδια μισά τμήματα στο χρόνο, τα οποία παραμένουν σχεδόν ίδια μετά τη διέλευσή τους από το κανάλι. Η κύρια διαφορά που έχουν ωφείλεται στο θόρυβο και στην ολίσθηση συχνότητας.²¹ Η εκτίμηση της αρχής του frame στο δέκτη, γίνεται με συνεχή αναζήτηση για ένα τέτοιο σύμβολο. Από τη στιγμή που βρεθεί, σταματά η αναζήτηση και αρχίζει η αποδιαμόρφωση των OFDM συμβόλων μέχρι το τέλος του frame. Κατόπιν, η διαδικασία επαναλαμβάνεται. Η μορφή του OFDM frame για το παρόν σύστημα παρουσιάζεται στο Σχ. 2.8.²²

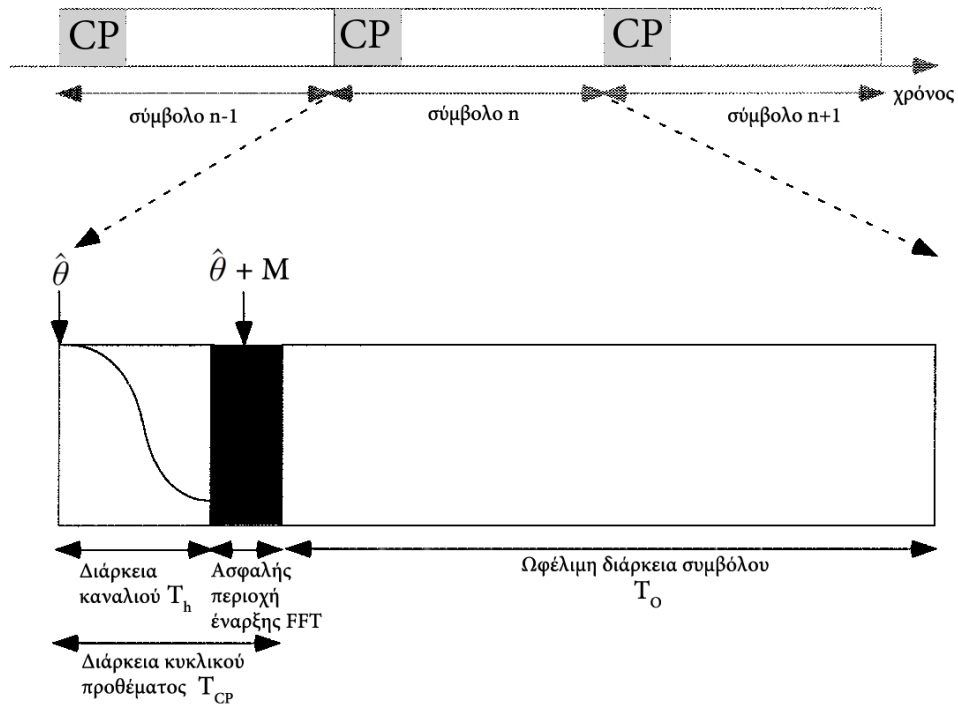
¹⁸Για μία αναλυτική περιγραφή των διαφορετικών τύπων συγχρονισμού, δείτε στο [LS06, ch. 4.2.1]

¹⁹Υπενθυμίζεται πως frame καλείται μία ομάδα από διαδοχικά OFDM σύμβολα.

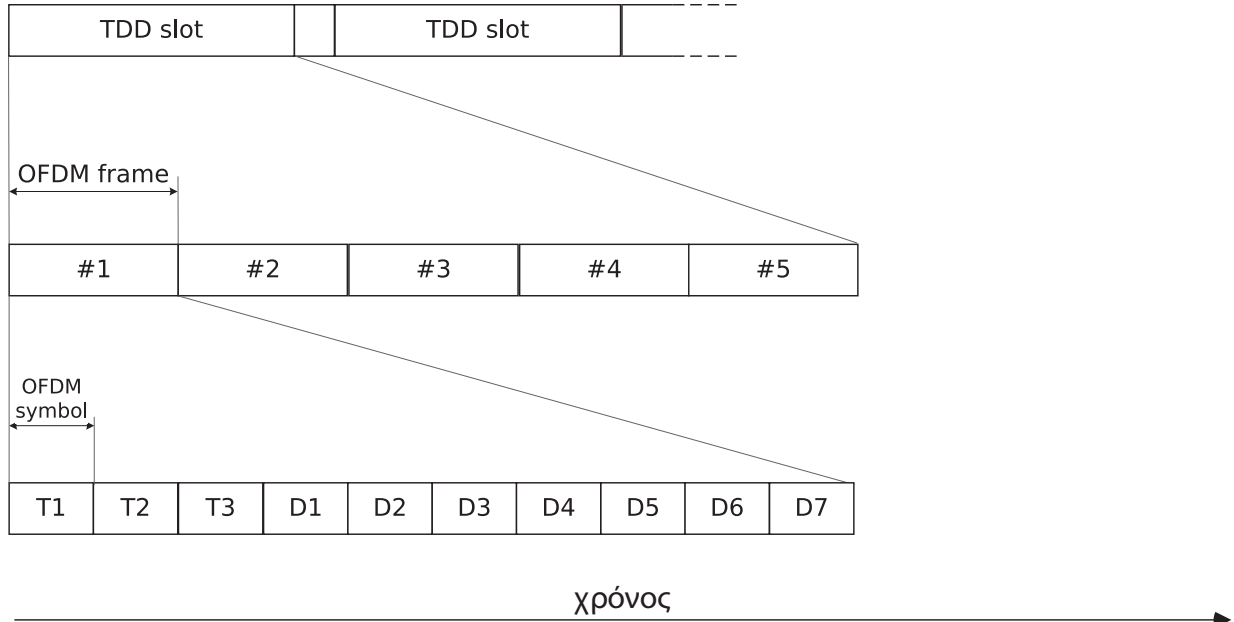
²⁰Η ύπαρξη του cyclic prefix προσδίδει στο σύστημα ανοχή σε σφάλματα συγχρονισμού.

²¹Για περαιτέρω συζήτηση πάνω στο θέμα της ολίσθησης συχνότητας, βλ. στην §2.2.4

²²Η ύπαρξη 3 training symbols θα αποσαφηνιστεί στις επόμενες παραγράφους.



Σχήμα 2.7: Περιθώριο σφάλματος συγχρονισμού



Σχήμα 2.8: OFDM frame

Επεξηγήσεις

Μορφή training symbol Το training symbol, που αποτελείται από δύο ίδια μισά τμήματα στο χρόνο, μπορεί να παραχθεί στη συχνότητα, διαμορφώνοντας μια ψευδοτυχαία ακολουθία δεδομένων στα άρτια

subcarrier και μία μηδενική ακολουθία στα περιττά ²³, συνήθως με QPSK. ²⁴ Αυτή η μορφή των training, τα κάνει να διαφέρουν από τα σύμβολα δεδομένων, καθώς αυτά είναι απίθανο να αποτελούνται από δύο ίδια μισά στο χρόνο. ²⁵

Μετρική Για την αναζήτηση του training symbol χρησιμοποιείται μία μετρική συνάρτηση που υπολογίζεται σε ένα παράθυρο (με μήκος όσο ένα training symbol) του σήματος στο χρόνο. Η μετρική υπολογίζει ένα μέγεθος που αποτελεί κάποια μορφή συνάρτησης αυτοσυσχέτισης (self-correlation) ανάμεσα στα περιεχόμενα δύο μισών του παραθύρου. Παρουσιάζει μέγιστο όταν το παράθυρο συμπίπτει με το training symbol, ενώ παρουσιάζει ελάχιστο όταν το παράθυρο αντιστοιχεί σε data symbol ή θόρυβο. Η εύρεση του σωστού σημείου συγχρονισμού, δηλαδή της αρχής του frame, προκύπτει από τη μεγιστοποίηση της μετρικής, καθώς αυτή υπολογίζεται σε ένα **ολισθαίνον παράθυρο** παρατήρησης.

Μοντέλο Δεδομένων των παρακάτω συμβολισμών :

$r(m)$: το λαμβανόμενο ψηφιοποιημένο μιγαδικό σήμα βασικής ζώνης

m : ο δείκτης διακριτού χρόνου

$L = N_o/2$: το μήκος του μισού ωφέλιμου συμβόλου,

d : ο δείκτης στο πρώτο sample ενός παραθύρου μεγέθους $2L$, καλείται και δείκτης παραθύρου,

ορίζονται τα εξής μεγέθη :

$$P(d) = \sum_{m=0}^{L-1} r^*(d+m) r(d+m+L)$$

$$R(d) = \sum_{m=0}^{L-1} |r(d+m+L)|^2$$

$$M(d) = \frac{|P(d)|^2}{R^2(d)}$$

όπου :

$M(d)$: η μετρική S&C που αντιστοιχεί στο παράθυρο d ,

$R(d)$: η ενέργεια του δευτέρου μισού του παραθύρου,

$P(d)$: ένα μιγαδικό άθροισμα γινομένων από ζεύγη samples που απέχουν L .

Είναι σαφές πως για να υπολογιστεί η τιμή της μετρικής στο πρώτο παράθυρο ($d = 0$), θα πρέπει να έχουν ληφθεί $2L$ samples. Αυτό μπορεί να εκφραστεί ορίζοντας $d = m - 2L$, δηλαδή ο δείκτης παραθύρου d έπεται του χρονικού δείκτη m κατά $2L$ samples.

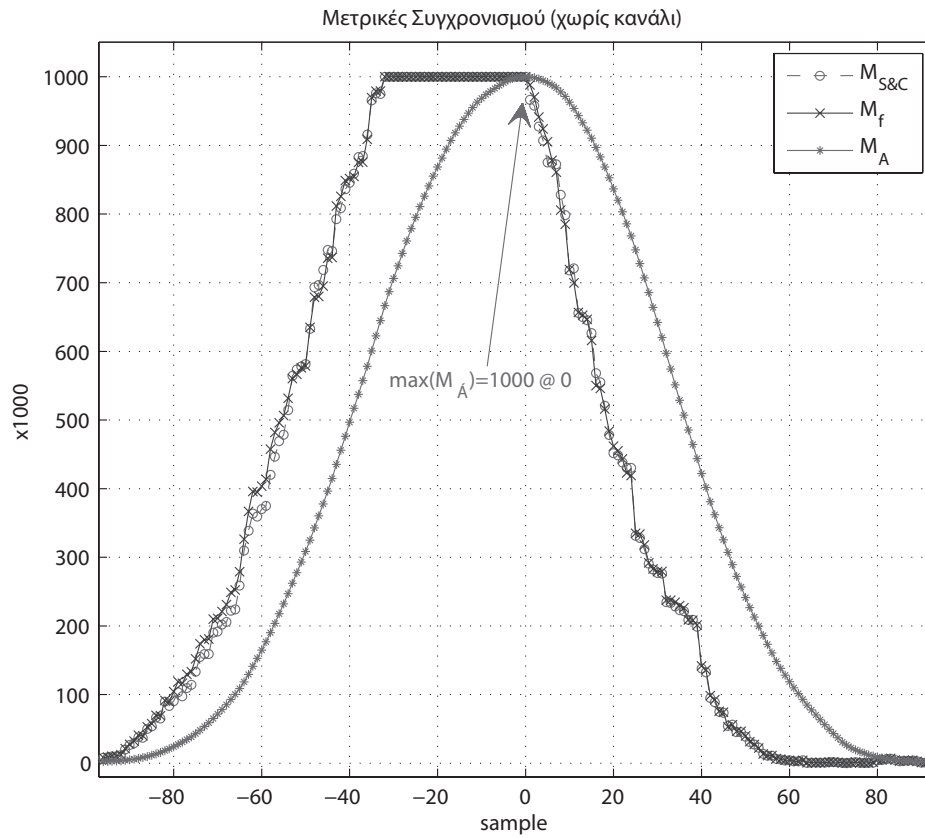
Τα όρια της τιμής της μετρικής είναι $M(d) \in [0 \dots 1]$, δηλαδή $\max_{\forall d} \{M(d)\} = 1$. Ωστόσο, όταν υπάρχει θόρυβος ή multipath στο κανάλι, η μέγιστη τιμή της μετρικής είναι μικρότερη του 1.

Πρόβλημα πλατό Ωστόσο, η παραπάνω μετρική, παρουσιάζει την εξής ιδιομορφία : Όπως φαίνεται στο σχήμα 2.9, υπάρχει ένα πλατό στη μετρική, κοντά στο σημείο συγχρονισμού, που προκύπτει από τη φύση του training symbol και της μετρικής. Το τέλος του (η δεξιά πλευρά του) συμπίπτει με την αρχή του ωφέλιμου συμβόλου. Το μήκος του (N_{PL}) εξαρτάται από το μήκος του καναλιού (N_h) και είναι $N_{PL} = N_{CP} - N_h + 1$. Στην περίπτωση όπου δεν εμφανίζονται φαινόμενα multipath στο κανάλι ή ισοδύναμα $N_h = 0$, τότε $N_{PL} = N_{CP} + 1$, δηλαδή το πλατό είναι ίσο με το μήκος του cyclic prefix. Όπως αναφέρθηκε

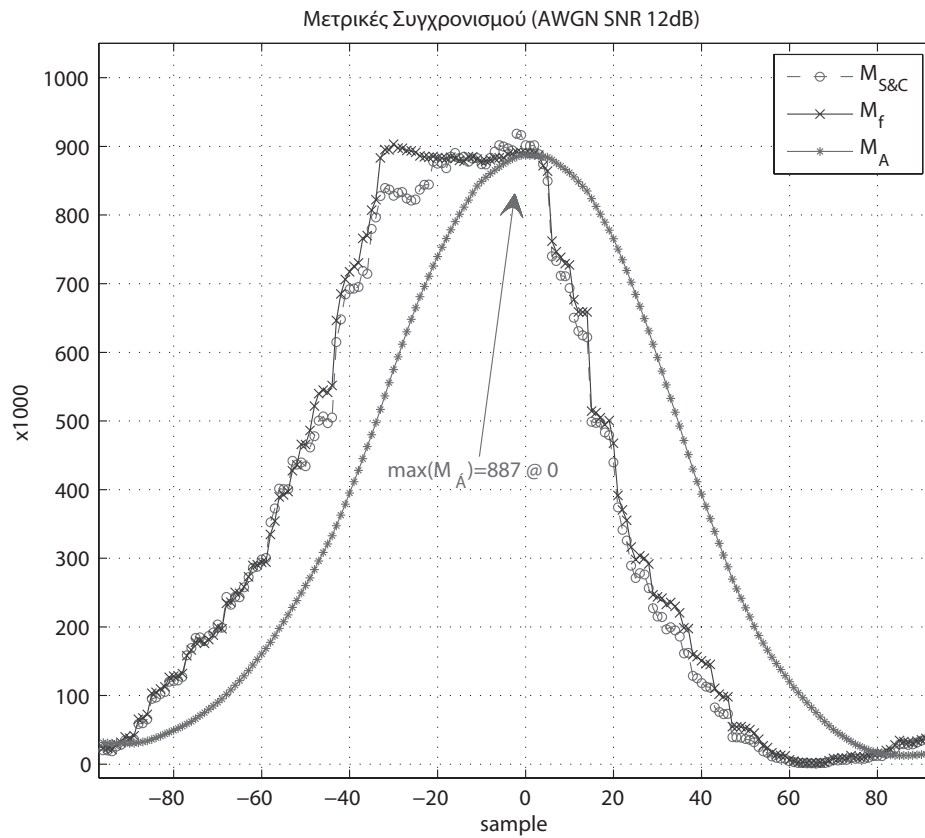
²³Αυτό προκύπτει από ιδιότητες του DFT.

²⁴Τα σημεία του αστερισμού QPSK, πολλαπλασιάζονται με $\sqrt{2}$ ώστε να διατηρηθεί η στάθμη ισχύος στα ίδια επίπεδα με τα data symbols.

²⁵Τα σύμβολα δεδομένων παράγονται στη συχνότητα, διαμορφώνοντας όλα τα subcarrier με μη μηδενικές ακολουθίες.



(α) Χωρίς θόρυβο



(β) Με θόρυβο AWGN SNR=12dB

Σχήμα 2.9: Μετρικές συγχρονισμού: S&C και Μέθοδος A

πιο πάνω, το σημείο συγχρονισμού μπορεί να επιλεγθεί ισοδύναμα από όλα τα σημεία του πλατό, αλλά το βέλτιστο σημείο συγχρονισμού είναι το μέσον του πλατό. Αυτό το σημείο πλεονεκτεί έναντι των υπολοίπων γιατί, τα σφάλματα συγχρονισμού που εμφανίζονται εντός μιας σχετικά μικρής περιοχής εκατέρωθεν του δεν οδηγούν σε ISI ή ICI. Η εύρεση αυτού του σημείου δεν είναι εύκολη διαδικασία, ιδιαίτερα σε κανάλια με θόρυβο. Ένας αλγόριθμος εύρεσης μεγίστου δεν αρκεί, καθώς το μέγιστο μπορεί να εμφανιστεί σε οποιοδήποτε σημείο του πλατό, αλλά και έξω από αυτό όταν υπάρχει θόρυβος. Έχει προταθεί στο [SC97] ο αλγόριθμος 90-Percent-Max²⁶ για την επίλυση αυτού του προβλήματος, αλλά η υλοποίησή του δεν είναι πρακτικά εφαρμόσιμη σε ένα σύστημα πραγματικού χρόνου.

Επίλυση με moving average Η μέθοδος A που προτείνεται, μεταξύ άλλων, στο [MZB00], είναι μία τροποποίηση της μετρικής S&C και εμφανίζει σαφώς καλύτερα ποιοτικά χαρακτηριστικά, με κύριο τη μη δημιουργία πλατό.

Συγκεκριμένα, η μετρική S&C τροποποιείται ως εξής:

- Εισάγεται στον παρονομαστή το μισό της ενέργειας ολόκληρου του παραθύρου, σε αντίθεση με την ενέργεια του δευτέρου μισού του παραθύρου.
- Εισάγεται ένα φίλτρο κινούμενου μέσου όρου (moving average filter) στην παλαιά μετρική με μήκος $N_{avg} = N_{CP} + 1$.

Το αποτέλεσμα είναι η απαλοιφή του πλατό και η παρουσία μεγίστου στο σημείο που αρχίζει το ωφέλιμο σύμβολο. Επίσης η μετρική είναι πιο εξομαλυσμένη, πράγμα που αφαιρεί ανεπιθύμητες κορυφές (spikes) λόγω θορύβου, όπως φαίνεται και στο σχήμα 2.9. Είναι φανερό το πλεονέκτημα της χρήσης της Μεθόδου A για την εύρεση του σημείου συγχρονισμού, καθώς τώρα μπορεί να χρησιμοποιηθεί ένας απλούστερος, σχετικά, αλγόριθμος εύρεσης μεγίστου. Τα όρια της τιμής της μετρικής είναι $M_A(d) \in [0 \dots 1]$, δηλαδή $\max_{\forall d} \{M_A(d)\} = 1$. Ωστόσο, όταν υπάρχει θόρυβος ή multipath στο κανάλι, η μέγιστη τιμή της μετρικής είναι μικρότερη του 1.²⁷

Μοντέλο (τροποποιημένο) Η τροποποιημένη μετρική $M_A(d)$ (Μέθοδος A) έχει ως εξής:

$$R_f(d) = \frac{1}{2} \sum_{m=0}^{2L-1} |r(d+m)|^2$$

$$M_f(d) = \frac{|P(d)|^2}{R_f^2(d)} \quad (2.1)$$

$$M_A(d) = \frac{1}{N_{CP} + 1} \sum_{k=-N_{CP}}^0 M_f(d+k)$$

όπου $R_f(d)$ είναι το ήμισυ της ενέργειας του παραθύρου.

Εύρεση μεγίστου Για την εύρεση του μεγίστου της μετρικής M_A , προτείνεται ο παρακάτω αλγόριθμος, που είναι κατάλληλος για λειτουργία σε πραγματικού χρόνου.²⁸ Η μετρική μεγιστοποιείται σε ένα **ολισθαίνων παράθυρο παρατήρησης**, εύρους M_{span} samples κάνοντας χρήση ενός κατωφλιού εγκυρότητας M_{rthres} . Δηλαδή το παράθυρο παρατήρησης ολισθαίνει στο χρόνο, όσο δεν έχει βρεθεί το μέγιστο. Το κατώφλι

$$M_{threshold} = \frac{M_{rthres}}{100} * \max_{\forall d} \{M_A(d)\} = \frac{M_{rthres}}{100}$$

²⁶Αφού υπολογιστεί η μέγιστη τιμή της μετρικής, βρίσκονται οι χρονικοί δείκτες της που έχουν τιμή 90% του μεγίστου. Το ημίαθροισμα αυτών των δεικτών αποτελεί την καλύτερη εκτίμηση για το μέσον του πλατό

²⁷Η τιμή του μεγίστου της μετρικής μπορεί να οδηγήσει σε μία πρώτη εκτίμηση για το SNR, όπως αναφέρεται στο [SC97].

²⁸Η κύρια ιδιομορφία που εισάγει η προδιαγραφή του πραγματικού χρόνου, είναι να μην χρειάζεται να σταματά η ροή των δεδομένων για να υπολογιστεί το αποτέλεσμα του αλγορίθμου.

αποτελεί μια ασφαλιστική δικλείδα, ελαχιστοποιώντας τις πιθανότητες απώλειας ή λανθασμένης ανίχνευσης κάποιου frame.²⁹ Το μήκος του παραθύρου παρατήρησης της μετρικής M_{span} , επιλέγεται έτσι ώστε να επιστραφεί ως αποτέλεσμα το ολικό μέγιστο και όχι κάποιο τοπικό μέγιστο που ενδεχομένως να παρουσιάζεται λόγω θορύβου. Δεδομένου ότι το μήκος της «καμπάνας» που παρουσιάζει η μετρική είναι N_o , τότε $N_o/2$ είναι η απόσταση του μεγίστου από το τέλος της «καμπάνας». Αυτή είναι η βέλτιστη τιμή που επιλέγεται για το M_{span} στην παρούσα υλοποίηση.

Διάστημα ασφαλείας Το μέγιστο που παρουσιάζει η μέθοδος Α βρίσκεται στην άκρη του παλαιού πλάτος, δηλαδή στο άκρο του διαστήματος ασφαλείας. Για να γίνει το σύστημα συγχρονισμού πιο ανθεκτικό σε σφάλματα χρονισμού, ο DFT εκκινεί από το κέντρο του πλάτος, δηλαδή από το βέλτιστο σημείο συγχρονισμού. Έστω d_{opt} το σημείο μεγιστοποίησης της M_A . Τότε το βέλτιστο σημείο συγχρονισμού είναι

$$(\hat{\theta} + M)_{opt} = d_{opt} - \frac{N_{CP} - N_h}{2} = d_{opt} - S_{offset}$$

Ορίζουμε το μέγεθος $S_{offset} = (N_{CP} - N_h)/2$, που είναι το offset της ανιχνευθείσας αρχής του frame.

Αναδρομικοί τύποι Επιπλέον όλοι οι παραπάνω τύποι έχουν και αναδρομική μορφή, δηλαδή είναι της μορφής $f(n+1) = f(n) + g(n)$, όπου $g(n)$ είναι συνάρτηση προηγούμενων τιμών της εισόδου. Σε realtime συστήματα, όπως το παρόν, οι αναδρομικοί τύποι είναι προτιμότεροι καθώς υλοποιούνται με την ελάχιστη δυνατή μνήμη, στον ελάχιστο δυνατό χρόνο. Οι αναδρομικοί τύποι για τις ανωτέρω μετρικές, λαμβάνοντας υπόψη ότι $m = d + 2L$, είναι:

$$P(d+1) = P(d) - r^*(m-2L)r(m-L) + r^*(m-L)r(m) \quad (2.2)$$

$$R_f(d+1) = R_f(d) - \frac{1}{2}|r(m-2L)|^2 + \frac{1}{2}|r(m)|^2 \quad (2.3)$$

$$M_A(d+1) = M_A(d) - \frac{M_f(d-N_{CP})}{N_{CP}+1} + \frac{M_f(d+1)}{N_{CP}+1} \quad (2.4)$$

όπου το $r(m)$ είναι το τρέχων sample, ενώ $M_A(d+1)$ είναι η τρέχουσα τιμή της μετρικής που αντιστοιχεί στο παράθυρο που αρχίζει με το sample $d+1 = m-2L+1$ και τελειώνει στο sample m , έχοντας μήκος $2L$.

Παράμετροι συγχρονισμού Το σύνολο των μεγεθών που έχουν οριστεί από τη μέχρι τώρα συζήτηση για το συγχρονισμό, συνοψίζονται στον πίνακα 2.3. Επίσης, αναγράφονται οι τιμές που έχουν επιλεγεί για την παρούσα σχεδίαση. Ωστόσο οι τιμές αυτές μπορούν να αλλάξουν εύκολα, όπως θα περιγραφεί στο Κεφάλαιο 3.

Κατώφλι εγκυρότητας της	M_{rthres}	60%
Εύρος αναζήτησης μετρικής	M_{span}	64 samples
Offset αρχής του frame	S_{offset}	8 samples

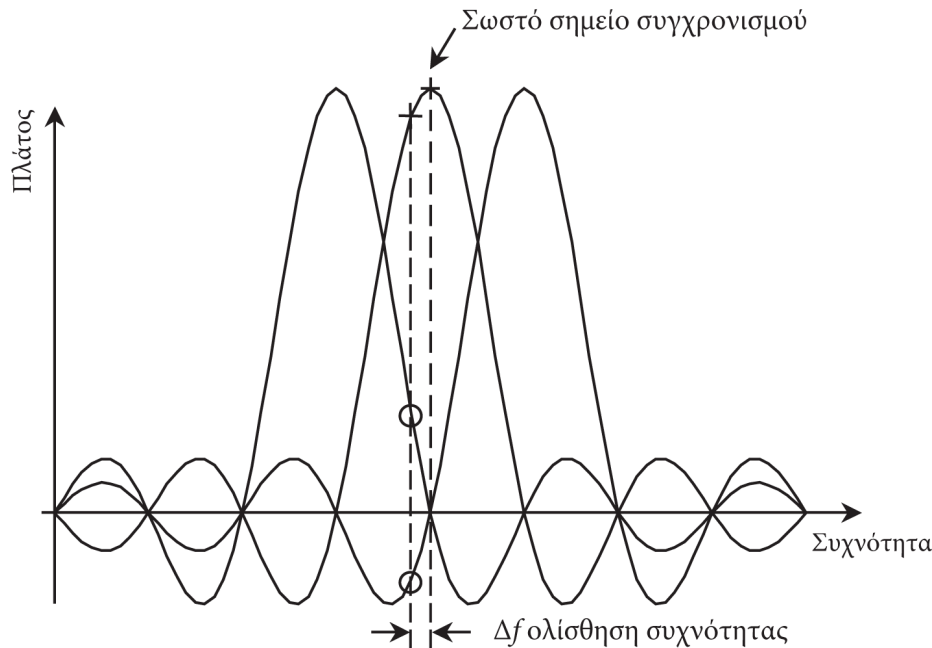
Πίνακας 2.3: Παράμετροι συγχρονισμού στο χρόνο

2.2.4 Συγχρονισμός στη Συχνότητα

Ο συγχρονισμός στη συχνότητα ή η εκτίμηση ολίσθησης συχνότητας (frequency offset estimation), είναι μία πολύ σημαντική διαδικασία για τα OFDM συστήματα. Τα τελευταία, ενώ παρουσιάζουν αυξημένη

²⁹Για μία συζήτηση πάνω στη βέλτιστη επιλογή του κατωφλιού, ο αναγνώστης παραπέμπεται στο [SC97].

ανοχή σε σφάλματα συγχρονισμού στο χρόνο, είναι πολύ ευαίσθητα σε σφάλματα ολίσθησης συχνότητας. Η ολίσθηση του φάσματος ενός OFDM σήματος οδηγεί σε απώλεια της ορθογωνιότητας μεταξύ των subcarrier με αποτέλεσμα την εμφάνιση ICI κατά την αποδιαμόρφωση (Σχ. 2.10). Αν το φάσμα παρουσιάζει ολίσθηση D_f , τότε προκαλείται στροφή φάσης στο OFDM σήμα ίση με $\Delta\phi = 2\pi t D_f$. Η μη αντιστάθμισης της στροφής φάσης, έχει ως αποτέλεσμα την περιστροφή του αστερισμού γύρω από την αρχή των αξόνων και τη διασπορά των σημείων του, όπως συμβαίνει με τον AWGN θόρυβο. Το OFDM παρουσιάζει πολύ μικρή ανοχή³⁰ σε σφάλματα ολίσθησης συχνότητας, συνεπώς ο συγχρονισμός συχνότητας πρέπει να γίνεται με μεγάλη ακρίβεια.



Σχήμα 2.10: Ολίσθηση συχνότητας subcarrier

Οι κύριοι μηχανισμοί που προκαλούν ολίσθηση συχνότητας είναι:

- α) τα φαινόμενα doppler λόγω σχετικής κίνησης πομπού και δέκτη, δηλαδή όταν το κανάλι θεωρείται χρονομεταβλητό,
- β) οι διαφορές στις συχνότητες των RF ταλαντωτών μεταξύ πομπού και δέκτη,
- γ) οι διαφορές στις συχνότητες δειγματοληψίας των DAC/ADC μεταξύ πομπού και δέκτη.

Τέτοιου είδους αποκλίσεις είναι αναμενόμενες σε πραγματικά συστήματα τηλεπικοινωνιών και πρέπει να διορθώνονται. Είναι φανερό από τα παραπάνω, πως ολίσθηση συχνότητας μπορεί να συμβεί και με στατικά κανάλια.

Αλγόριθμος

Όπως αναφέρεται στον αλγόριθμο S&C στο [SC97], η ολίσθηση συχνότητας μπορεί να εκτιμηθεί από το training symbol που έχει χρησιμοποιηθεί για το συγχρονισμό στο χρόνο. Όταν το training symbol περάσει από το κανάλι, το πρώτο μισό θα είναι παρόμοιο με το δεύτερο, εκτός από μία διαφορά φάσης που προέρχεται από την ολίσθηση συχνότητας. Ο πολλαπλασιασμός του μιγαδικού συζυγούς ενός sample του πρώτου μισού με το αντίστοιχο sample του δεύτερου μισού, $T_o/2$ δευτερόλεπτα αργότερα, θα έχει φάση $\phi = \pi t D_f$, που είναι ίση με τη διαφορά φάσης των δύο sample. Η μετρική $P(d)$ που αναφέρθηκε πιο

³⁰της τάξης ενός κλάσματος του subcarrier spacing

πάνω, είναι ένα άθροισμα τέτοιων γινομένων.³¹ Στην αρχή του frame, όταν το παράθυρο της μετρικής βρίσκεται κοντά training symbol, τα γινόμενα αυτά έχουν περίπου την ίδια φάση, άρα και το άθροισμα τους έχει την ίδια φάση. Άλλωστε αυτός είναι και ο λόγος που το πλάτος της μετρικής, μεγιστοποιείται σε αυτό το σημείο. Συνεπώς η καλύτερη εκτίμηση της διαφοράς φάσης μεταξύ των δύο μισών του training symbol, θα είναι

$$\hat{\phi} = \angle P(d_{opt})$$

δηλαδή η φάση της μετρικής $P(d)$ στο σημείο συγχρονισμού.

Coarse frequency estimation Έχοντας εκτιμήσει την $\hat{\phi}$, αποκτάται η χονδρική εκτίμηση της ολίσθησης συχνότητας (coarse frequency estimation) ως:

$$\hat{\psi}_f = \frac{\hat{\phi}}{\pi T_o}$$

Αντιστάθμιση Εάν για τη διαφορά φάσης ισχύει $|\phi| < \pi$ ή ισοδύναμα για την ολίσθηση συχνότητας ισχύει $|D_f| < \Delta_f$,³² η τελική εκτίμηση είναι $\hat{D}_f = \hat{\psi}_f$. Συνεπώς τα επόμενα OFDM σύμβολα, μετά το training symbol, αντισταθμίζονται ως εξής (στο χρόνο)

$$r_c(t) = r(t) \exp(-j2\pi t \hat{D}_f) = r(t) \exp(-j \frac{2t\hat{\phi}}{T_o})$$

Fine frequency estimation Στην αντίθετη περίπτωση που η ολίσθηση συχνότητας είναι μεγαλύτερη από το subcarrier spacing, θα πρέπει να γίνει και ραφινάτη εκτίμηση (fine frequency estimation). Η διαδικασία για την εύρεση της fine εκτίμησης Ω_f , περιλαμβάνει την μετάδοση ενός δεύτερου training symbol και τη μεγιστοποίηση μιας μετρικής στο πεδίο της συχνότητας. Σε αυτή την περίπτωση η συνολική εκτίμηση της ολίσθησης συχνότητας είναι το άθροισμα της coarse και fine εκτίμησης³³, $\hat{D}_f = \psi_f + \Omega_f$. Ωστόσο δε θα γίνει περαιτέρω αναφορά στο παρόν³⁴, καθώς το fine frequency estimation είναι εκτός των προδιαγραφών του συστήματος που υλοποιείται, όπως έχουν τεθεί στην §2.2.1. Στο παρόν σύστημα θεωρείται πως $\Omega_f = 0$.

2.2.5 Εκτίμηση Καναλιού

Η εκτίμηση του καναλιού (channel estimation) συνίσταται στην προσπάθεια ανακατασκευής της συνάρτησης μεταφοράς του καναλιού H . Εν γένει, σε ένα χρονομεταβλητό multipath κανάλι, η συνάρτηση μεταφοράς είναι μία 2Δ συνάρτηση, του χρόνου και της συχνότητας. Η εκτίμησή της γίνεται με την εισαγωγή πιλοτικών τόνων (pilot tones) στο δισδιάστατο πλέγμα χρόνου-συχνότητας. Για κάθε διάταξη των πιλοτικών τόνων, υπάρχουν διαφορετικοί αλγόριθμοι εκτίμησης, που διακρίνονται ανάλογα με την επίδοση και την πολυπλοκότητά τους. Η βέλτιστη εκτίμηση είναι με 2Δ παρεμβολή, πλην όμως η πολυπλοκότητά της είναι απαγορευτική για πρακτικές υλοποιήσεις. Ωστόσο μπορεί να επιτευχθεί ικανοποιητική εκτίμηση με 1Δ channel estimators.

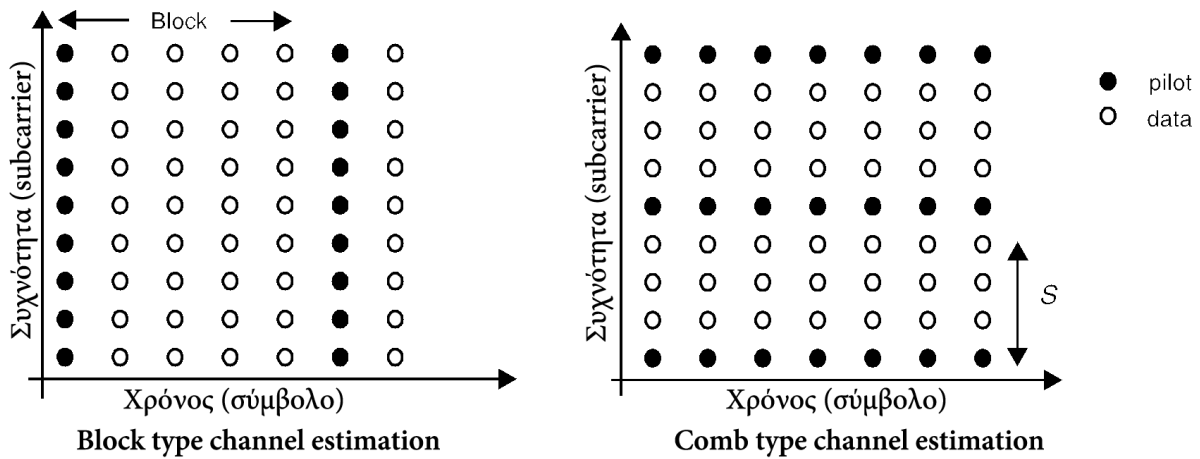
³¹ Η $P(d)$ επαναλαμβάνεται εδώ

$$P(d) = \sum_{m=0}^{L-1} r^*(d+m)r(d+m+L)$$

³² Δηλαδή εάν η ολίσθηση συχνότητας είναι μικρότερη από το subcarrier spacing

³³ Ορίζοντας το μέγεθος της κανονικοποιημένης ολίσθησης συχνότητας ως προς το subcarrier spacing $v_f = D_f/\Delta_f$, το D_f εκφράζεται σε μονάδες "subcarrier spacing". Για παράδειγμα μπορεί να ειπωθεί ότι η ολίσθηση συχνότητας είναι 1,3 subcarrier spacings. Με αυτόν τον τρόπο είναι δυνατή η ερμηνεία του coarse frequency estimation ως το κλασματικό μέρος του v_f , ενώ του fine frequency estimation ως το ακέραιο.

³⁴ Η μέθοδος εύρεσης του fine frequency estimation περιγράφεται στο [SC97]

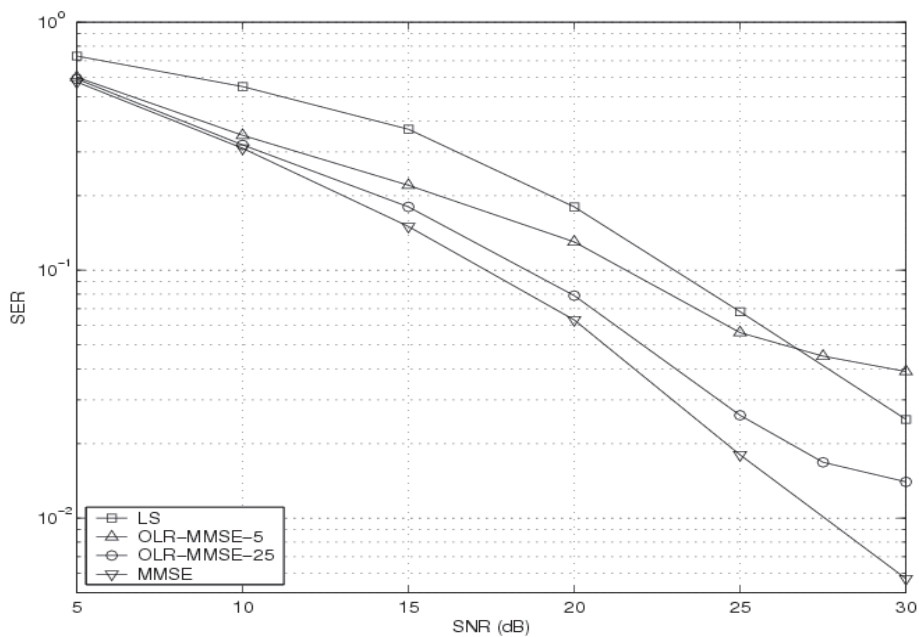


Σχήμα 2.11: Τύποι αλγορίθμων εκτίμησης καναλιού

Οι δύο τύποι των 1Δ channel estimators είναι ο τύπου μπλοκ (block type) και ο τύπου χτένας (comb type) (Σχ. 2.11). Στους comb type, εισάγονται πιλοτικοί τόνοι σε συγκεκριμένα subcarrier, σε κάθε OFDM σύμβολο που μεταδίδεται. Έχουν τη δυνατότητα να παρακολουθούν γρήγορες αλλαγές στο κανάλι και είναι κατάλληλοι για fast fading κανάλια.

Στους block type, μεταδίδονται ανα τακτά χρονικά διαστήματα, σύμβολα με πιλοτικούς τόνους σε όλα τα subcarrier (training blocks ή symbols), τα οποία δε φέρουν payload. Ο δέκτης, έχοντας ως γνωστό το training block και ενδεχομένως κάποια στατιστικά μεγέθη του καναλιού, πραγματοποιεί την εκτίμηση του καναλιού. Οι block type αλγόριθμοι, δεν μπορούν να παρακολουθούν γρήγορες αλλαγές στο κανάλι και είναι κατάλληλοι για στατικά ή πολύ αργά μεταβαλλόμενα κανάλια.

Αλγόριθμος



Σχήμα 2.12: Σύγκριση LS, MMSE αλγορίθμων εκτίμησης καναλιού

Το παρόν σύστημα, όπως προκύπτει και από τις προδιαγραφές στην §2.2.1, σχεδιάζεται για στατικά ή πολύ

αργά μεταβαλλόμενα κανάλια (της τάξης ενός OFDM frame), συνεπώς έχει επιλεχθεί ένας block type αλγόριθμος.

Συγκεκριμένα υλοποιείται ο αλγόριθμος ελαχίστων τετραγώνων (LS, Least Squares). Το πλεονέκτημα που παρουσιάζει ο LS είναι η μικρή υπολογιστική πολυπλοκότητα και η απλότητα στην υλοποίηση. Ωστόσο η απόδοσή του δεν είναι η καλύτερη συγκριτικά με άλλους block based αλγορίθμους, όπως ο *Ελάχιστου Μέσου Τετραγωνικού Σφάλματος (MMSE, Minimum Mean Square Error)*. Όπως φαίνεται στο Σχήμα 2.12 η απόδοση του LS είναι η χειρότερη, ενώ του MMSE η καλύτερη. Το αντίστροφο ισχύει για τις πολυπλοκότητές τους. Οι γραμμές OLR-MMSE-5 και OLR-MMSE-25, αναφέρονται σε μία απλοποιημένη εκδοχή του MMSE (*Optimal Low Rank - OLR*), που είναι κατάλληλη για πρακτική υλοποίηση. Για μία εκτενέστερη παρουσίαση των αλγορίθμων εκτίμησης καναλιού, ο αναγνώστης παραπέμπεται στο [SM06].

Η διαδικασία της εκτίμησης καναλιού, πραγματοποιείται μία φορά σε κάθε frame μετά από τη διαδικασία του συγχρονισμού. Για να γίνει αυτό εισάγεται ένα training block (symbol) στην αρχή κάθε frame, μετά από τα training symbols που χρησιμοποιούνται για τη διαδικασία του συγχρονισμού. Μόλις ολοκληρωθεί η εκτίμηση, αρχίζει η διαδικασία της ισοστάθμισης καναλιού (channel equalization), αφού τα σύμβολα που ακολουθούν περιέχουν μόνο ωφέλιμη πληροφορία (data symbols). Όλα τα data symbols, μέχρι το πέρας του frame ισοσταθμίζονται με την ίδια εκτίμηση καναλιού. Αυτό είναι δυνατόν εφόσον το σύστημα χρησιμοποιείται σε πολύ αργά μεταβαλλόμενα κανάλια, πρακτικά στατικά για τη διάρκεια ενός frame. Η διάρκεια του frame επιλέγεται έτσι ώστε να ικανοποιείται αυτός ο περιορισμός, ελαχιστοποιώντας ταυτόχρονα την επιβάρυνση που εισάγεται από το προοίμιο του frame.

Μοντέλο καναλιού Έστω ότι:

$X(k)$ είναι ένα μεταδιδόμενο OFDM σύμβολο στο πεδίο της συχνότητας

$Y(k)$ είναι το αντίστοιχο λαμβανόμενο σύμβολο, στο πεδίο της συχνότητας, μετά τη διέλευσή του από το κανάλι.

$H(k)$ είναι ο DFT της κρουστικής απόκρισης του καναλιού, δηλαδή η συνάρτηση μεταφοράς του καναλιού $h(t)$, όπως έχει δειγματοληπτηθεί στο δέκτη,

$N(k)$ είναι ο DFT της στοχαστικής διαδικασίας λευκού θορύβου (AWGN), $n(t)$

Το ληφθέν σήμα στο δέκτη είναι $y(t) = h(t) \otimes x(t) + n(t)$ ή ισοδύναμα στο πεδίο της συχνότητας $Y(k) = H(k) X(k) + N(k)$.

Εκτίμηση καναλιού Έστω ακόμα ότι:

$X_T(k)$ είναι το μεταδιδόμενο training symbol στο πεδίο της συχνότητας, το οποίο είναι γνωστό και στο δέκτη,

$Y_T(k)$ είναι το αντίστοιχο λαμβανόμενο training symbol, στο πεδίο της συχνότητας, μετά τη διέλευσή του από το κανάλι,

τότε η εκτίμηση του καναλιού σύμφωνα με τον αλγόριθμο LS είναι

$$\hat{H}_{LS}(k) = Y_T(k)/X_T(k) \quad k \in 0, \dots, N - 1$$

Ισοστάθμιση καναλιού Όπως αναφέρθηκε στην §2.1.4, η ισοστάθμιση σε ένα OFDM σύστημα είναι μία απλή πράξη μιγαδικής διαίρεσης του κάθε subsymbol με την αντίστοιχη εκτίμηση καναλιού. Εάν $X_{RX}(k)$ είναι ένα ισοσταθμισμένο OFDM σύμβολο στο πεδίο της συχνότητας ισχύει:

$$X_{RX}(k) = \frac{Y(k)}{\hat{H}(k)} \quad k \in 0, \dots, N - 1$$

Κεφάλαιο 3

Το Σύστημα *per se*

3.1 Εξωτερική Όψη του Συστήματος

3.1.1 Βασικά Χαρακτηριστικά

Αν ζητούσε κανείς μια λακωνική περιγραφή του συστήματος σε μια φράση, αυτή θα ήταν κάπως έτσι:

“Πρόκειται για έναν realtime ασύρματο baseband OFDM πομποδέκτη, ικανό να λειτουργήσει με simplex ή half duplex τρόπο μετάδοσης, υλοποιημένο σε FPGA.”

Δεδομένου ότι στα κεφάλαια 1 και 2 έχουν αναλυθεί οι περισσότεροι από τους παραπάνω όρους, εξήγησής χρήζει η έννοια του συστήματος πραγματικού χρόνου (*real-time system*).

Συστήματα Πραγματικού Χρόνου (*Real-Time Systems*)

Με δυο λόγια, ένα σύστημα *real-time* είναι ένα σύστημα το οποίο υπόκειται σε χρονικούς περιορισμούς, δηλαδή υπάρχουν προθεσμίες (*deadlines*) προκειμένου να εκτελεστούν οι λειτουργίες που αυτό επιτελεί.¹ Τα συστήματα πραγματικού χρόνου λειτουργούν υπο καθεστώς πολύ αυστηρών περιορισμών οσον αφορά στο χρόνο απόκρισής τους.²

Πιο συγκεκριμένα, τώρα, προκειμένου ένα σύστημα να χαρακτηριστεί ως *real-time*, θα πρέπει η ολική ορθότητα μιας λειτουργίας να εξαρτάται όχι μόνο από τη λογική της ορθότητα, αλλά και από το χρόνο που χρειάζεται να τελεσφορήσει. Τα *real time* συστήματα κατατάσσονται σε τρεις κατηγορίες, ανάλογα με τις επιπτώσεις της μη τήρησης ενός *deadline*. Ο πίνακας 3.1 παρουσιάζει αυτήν την κατάταξη.

Σύμφωνα με τα παραπάνω, το υλοποιηθέν σύστημα εμπίπτει, σαφώς, στην κατηγορία των *hard real-time* συστημάτων. Τόσο στον πομπό όσο και στο δέκτη (κυρίως στην υπομονάδα συγχρονισμού), οποιαδήποτε εκπνοή κάποιας χρονικής προθεσμίας (*deadline*) οδηγεί σε μη αναστρέψιμη απώλεια δεδομένων, γεγονός απαράδεκτο για ένα τηλεπικοινωνιακό σύστημα. Αυτή ακριβώς η απαίτηση έχει καθοδηγήσει τη σχεδίαση των μονάδων του συστήματος, όπου πολλές φορές ο χώρος θυσιάστηκε προς όφελος της ταχύτητας στο βαθμό που αυτή καθορίζει την συμμόρφωση (*compliance*) με τους αυστηρούς χρονικούς περιορισμούς.

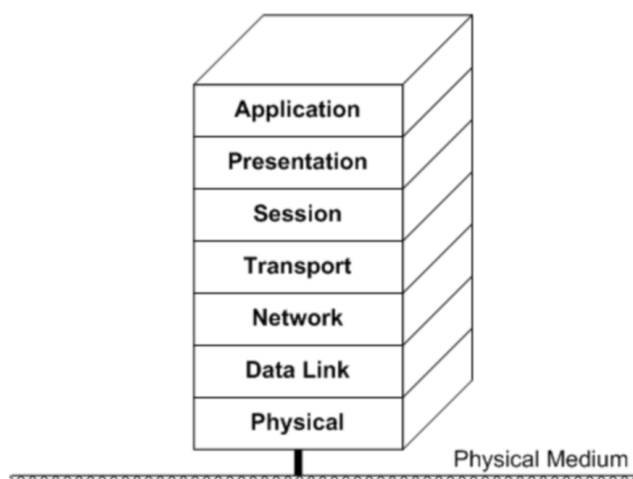
Προχωρώντας λίγο παραπέρα τη συζήτηση περι *real-time*, το σύστημα κατατάσσεται στο Φυσικό Επίπεδο

¹Σε αντιδιαστολή, ένα σύστημα *μη-πραγματικού χρόνου* (*non-real-time system*) είναι αυτό στο οποίο δεν υπάρχουν *deadlines*, ακόμα κι αν η γρήγορη απόκριση ή η υψηλή απόδοση είναι επιθυμητές.

²Εδώ θα πρέπει να σημειωθεί ότι ο όρος *real-time* είναι εξ'ολοκλήρου ανεξάρτητος από την έννοια της ταχύτητας επεξεργασίας. Πιο συγκεκριμένα, ικανή και αναγκαία συνθήκη προκειμένου ένα σύστημα να ονομαστεί *real-time* είναι, αποκλειστικά, η ικανοποίηση των χρονικών περιορισμών που του έχουν τεθεί. Το αν αυτοί επιβάλλουν ταχύτερη ή πιο χαλαρή επεξεργασία δεν έχει σχέση με τον χαρακτηρισμό *real-time*.

Hard	Η εκπνοή προθεσμίας οδηγεί σε ολική αστοχία συστήματος
Firm	Σπάνιες εκπνοές είναι ανεκτές, αλλά με ενδεχόμενη υποβάθμιση του QoS. Η χρησιμότητα ενός αποτελέσματος είναι μηδενική μετά την εκπνοή της προθεσμίας του
Soft	Η χρησιμότητα ενός αποτελέσματος μειώνεται μετά το πέρας της εκπνοής και αυτό αντανάκλαται στο QoS του συστήματος

Πίνακας 3.1: Κατάταξη real-time συστημάτων



Σχήμα 3.1: Το μοντέλο OSI

(*Physical Layer*) του μοντέλου OSI.³ Αν, τώρα, το σύστημα θεωρηθεί ως μέρος ενός ευρύτερου συστήματος επικοινωνίας (π.χ. *IEEE 802.11*) τότε τα επίπεδα *Data Link* και *Network* (σχήμα 3.1) αναλαμβάνουν την αποκατάσταση διαφόρων σφαλμάτων και τη διατήρηση του επιθυμητού QoS.⁴ Έτσι θεωρούμενο ομού, το (συνολικό) σύστημα εμπίπτει, ενδεχομένως, στις κατηγορίες *firm* ή/και *soft*. Παρ'όλα αυτά, ο πομποδέκτης μόνος του ανήκει, όπως προαναφέρθηκε, στα *hard real-time* συστήματα.

Γενικότερα Ζητήματα Χρονισμού

Σε έναν χονδροειδή παραλληλισμό με το ανθρώπινο σώμα, θα μπορούσε κάποιος να πει ότι, αν ο αλγόριθμος που υπαγορεύει τη λειτουργία του συστήματος είναι ο εγκέφαλος και αν οι διαφορές υπομονάδες του συστήματος είναι τα όργανα του σώματος τότε σίγουρα το ρολόι του συστήματος είναι η καρδιά. Πράγματι, στην ηλεκτρονική και κυρίως στα σύγχρονα (*synchronous*) ψηφιακά συστήματα, το σήμα⁵ ρολογιού (*clock signal*) είναι ένα σήμα που ταλαντώνεται ανάμεσα σε δύο καταστάσεις (*high*, *low*) και χρησιμοποιείται εν είδει μετρονόμου για το συντονισμό των ενεργειών και λειτουργιών των διάφορων υπομονάδων του συστήματος.

Για την παραγωγή αυτού του σήματος ρολογιού χρησιμοποιείται ένας ειδικός τύπος κυκλώματος ο οποίος ονομάζεται *γεννήτρια ρολογιού*. Κοινό παρονομαστή κάθε τύπου γεννήτριας ρολογιού αποτελούν το κύκλωμα συντονισμού (*resonant circuit*) και το κύκλωμα ενίσχυσης (*amplifier circuit*).⁶

³Open Systems Interconnection

⁴Quality Of Service

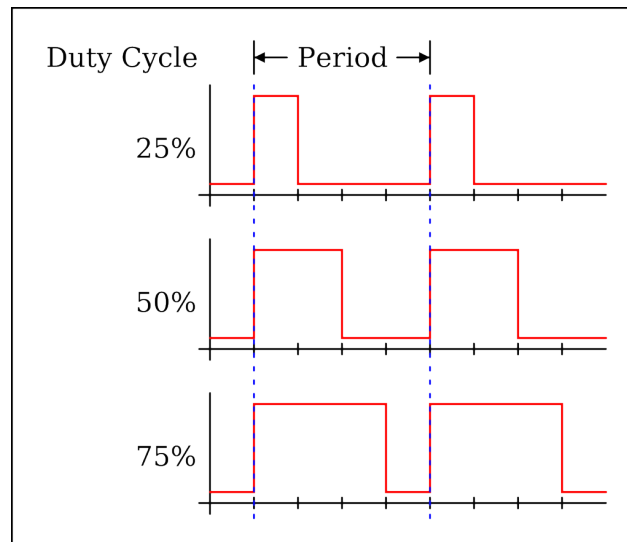
⁵Η έννοια του όρου *σήμα* μπορεί να αποδοθεί ως *μεταδιδόμενη ενέργεια φέρουσα πληροφορία*.

⁶Το κύκλωμα συντονισμού αποτελείται συνήθως από ένα *πιεζοηλεκτρικό ταλαντωτή χαλαζία* (κρύσταλλο) αν και, ορισμένες φορές, χρησιμοποιούνται πιο απλά κυκλώματα (όπως *RC* και *LC*). Το κύκλωμα ενίσχυσης αντιστρέφει την έξοδο του ταλαντωτή και επανατροφοδοτεί μέρος αυτού του σήματος πίσω στον ταλαντωτή προκειμένου να διατηρηθεί η ταλάντωση.

Σε αυτό το σημείο είναι χρήσιμο να οριστεί η έννοια του ρυθμού ρολογιού (*clock rate*). Ο ρυθμός ρολογιού είναι ο αριθμός των κύκλων ανά δευτερόλεπτο (μετρημένος σε *Hz*) ή η συχνότητα ταλάντωσης του ρολογιού. Ένας κύκλος ρολογιού (*clock cycle*) είναι ο χρόνος που απαιτείται για μια πλήρη ταλάντωση του σήματος ρολογιού. Σχετικό με τα προηγούμενα είναι και ένα μέγεθος γνωστό ως *duty cycle*, το οποίο μπορεί να οριστεί ως "ο χρόνος κατά τον οποίο το σήμα ρολογιού είναι ενεργό (συνήθως high ή στάθμης '1') ως κλάσμα του συνολικού χρόνου *clock cycle* T_{cc} ", όπου $T_{cc} = T_{high} + T_{low}$ δίνοντας επομένως την ακόλουθη σχέση για το *duty cycle*:⁷

$$Duty = \frac{T_{high}}{T_{cc}}$$

Το σχήμα 3.2 απεικονίζει σήματα ρολογιού (τετραγωνικούς παλμούς) με διάφορες τιμές για το *duty cycle*.



Σχήμα 3.2: Διάφορες τιμές για το *duty cycle*

Το παραγόμενο σήμα ρολογιού έχει μορφή η οποία κυμαίνεται από μια απλή συμμετρική τετραγωνική κυματομορφή έως πιο σύνθετες δομές (*arrangements*). Πάντως, η πιο συνηθισμένη μορφή του είναι αυτή της τετραγωνικής παλμοσειράς με *duty cycle* 50% και με σταθερή, συγκεκριμένη συχνότητα.

Ρολόι Συστήματος

Στον πομποδέκτη OFDM το κεντρικό ρολόι συστήματος είναι χρονισμένο στα X MHz και έχει τη μορφή τετραγωνικής παλμοσειράς με *duty cycle* 50%. Παράλληλα με το κεντρικό ρολόι, το σύστημα τροφοδοτείται και με ένα δεύτερο ρολόι χρονισμένο στα Y MHz. Επομένως μπορεί να ειπωθεί ότι ο πομποδέκτης είναι ένα *dual-clock* σύστημα. Ο όρος *dual clock* αναφέρεται σε ένα σύστημα με δύο ανεξάρτητα ρολόγια. Το δεύτερο ρολόι μπορεί να παράγεται από έναν δεύτερο ταλαντωτή ή από ένα *Βρόχο Κλειδωμένης Φάσης* (*Phase Locked Loop* εφεξής PLL). Η ύπαρξη του δεύτερου ρολογιού επιτρέπει σε κάποια υποσυστήματα του κυκλώματος να εργάζονται με μεγαλύτερη (ή μικρότερη) ταχύτητα χρονισμού από τα υπόλοιπα. Το σύνολο των υποσυστημάτων που εργάζονται υπό το ίδιο ρολόι, αποτελεί μία *περιοχή χρονισμού* (*clock domain*).

Οι λόγοι που μπορεί να είναι χρήσιμη αυτή η τεχνική, είναι η υλοποίηση απαιτητικών αλγορίθμων απλωμένων στο χρόνο (π.χ. FFT) ή η αλλαγή της ροής των δεδομένων (π.χ. έξοδος 1 έναντι 4 δεδομένων εισόδου). Βέβαια, θα μπορούσαν να εργάζονται όλα τα υποσυστήματα με το ταχύτερο ρολόι που υπάρχει στο σύστημα. Αυτό, όμως, θα είχε ως συνέπεια την εισαγωγή αυστηρότερων περιορισμών χρονισμού καθιστώντας όλη τη διαδικασία ανάλυσης χρονισμού (*timing analysis*) εξαιρετικά επίπονη. Εκτός αυτού η αύξηση της ταχύτητας του ρολογιού, οδηγεί και σε αύξηση της κατανάλωσης ισχύος.

⁷Πιο σωστά, βέβαια, αποδίδεται από την έκφραση $\frac{T_{active}}{T_{active}+T_{inactive}} = \frac{T_{active}}{T_{cc}}$

Ένα θέμα που πρέπει να προσεχτεί σε dual clock συστήματα, είναι η διάσχιση⁸ των clock domain. Για την αποφυγή αυτού του φαινομένου (ουσιαστικά για τη μείωση κατά πολύ της πιθανότητας εμφάνισής του) χρησιμοποιούνται ειδικές διατάξεις που ονομάζονται *συγχρονιστές*. Ωστόσο όταν πρέπει να μεταφερθούν σήματα με εύρος μεγαλύτερο του ενός bit, οι συγχρονιστές δεν αρκούν και χρησιμοποιούνται άλλες διατάξεις. Μία από αυτές είναι το dual clock FIFO (First-In First-Out Buffer). Συμπεριφέρεται όπως ένα single clock FIFO, πλην όμως η είσοδος και η έξοδος του είναι χρονισμένες σε διαφορετικά ρολόγια.

Αναπαράσταση αριθμών

Ο κύριος τύπος αναπαράστασης αριθμών στο σύστημα είναι ο *προσημασμένος ακέραιος* (signed integer) εύρους 16 bit (int16). Αυτό σημαίνει ότι τα αριθμητικά σήματα εισόδου και εξόδου του top-level αλλά και τα περισσότερα ενδιάμεσα μεταξύ των μονάδων έχουν εύρος 16 bit.⁹

Ως γνωστόν, το δυναμικό εύρος¹⁰ του τύπου int16 είναι το διάστημα

$$[\min_int16, \max_int16] = [-32768, 32767]$$

Όταν ένας αριθμός είναι εκτός του παραπάνω εύρους, μπορεί να υποστεί *κορεσμό* (saturation), δηλαδή ο τελικός αριθμός που θα αποθηκευτεί, να τεθεί στον μέγιστο ή τον ελάχιστο αναπαραστάσιμο ακέραιο. Η άλλη εναλλακτική είναι η *αποκοπή* (truncation), όπου κόβονται τα ψηφία που περισσεύουν. Το αποτέλεσμα του κορεσμού σε ένα ψηφιακό σήμα (samples), είναι το *ψαλίδισμα* των κορυφών του (clipping). Αν και το clipping εισάγει παραμόρφωση, είναι προτιμότερο από την αποκοπή, που μπορεί να οδηγήσει σε τυχαία επιλογή τελικού αριθμού.

3.1.2 Περιγραφή Ακροδεκτών Εισόδου/Εξόδου

Στο σχήμα 3.3 απεικονίζεται το υψηλότερο επίπεδο (top level) του συστήματος. Αν υλοποιηθεί σε ένα IC, αυτή θα είναι η εξωτερική του εμφάνιση. Διακρίνεται ένα πλήθος ακροδεκτών οι οποίοι, χωροταξικά, έχουν ως εξής : Αριστερά βρίσκονται οι εισοδοί του συστήματος και δεξιά οι έξοδοι, ενώ το σχέδιο δίνεται σε μονογραμμική μορφή. Οι ακροδέκτες ομαδοποιούνται ανά λειτουργία πομπού ή δέκτη (TX, RX αντίστοιχα).

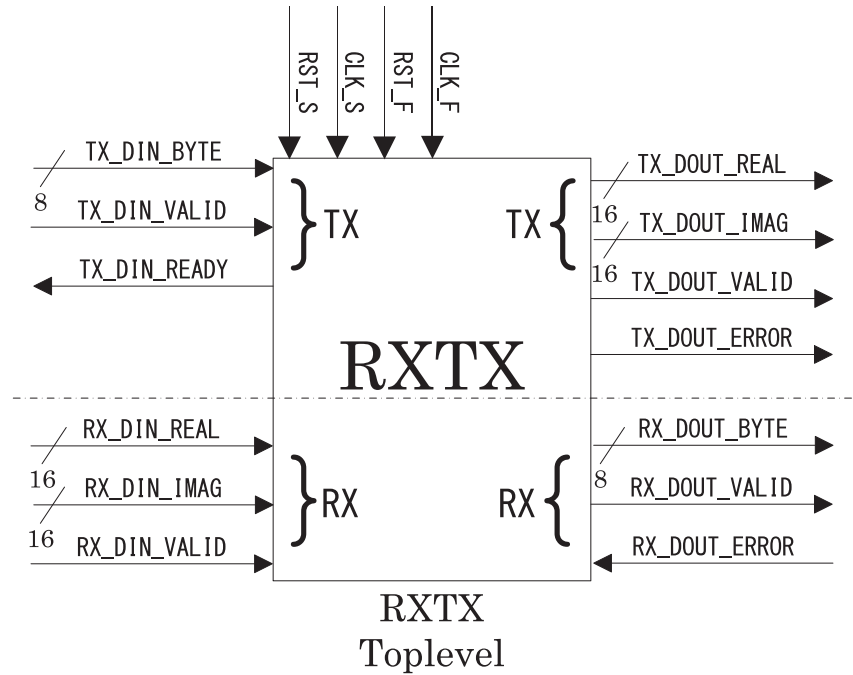
Αναλυτικότερα, η κατηγοριοποίηση των ακροδεκτών έχει ως εξής :

- Κοινοί Ακροδέκτες
 - Είσοδοι : { clk_s, clk_f, rst_s_n, rst_f_n }
 - Έξοδοι : { ∅ }
- Πομπός (TX)
 - Είσοδοι : { tx_din_valid, tx_din_byte }
 - Έξοδοι : { tx_dout_valid, tx_dout_real, tx_dout_imag, tx_din_ready, tx_dout_error }
- Δέκτης (RX)
 - Είσοδοι : { rx_din_valid, rx_din_real, rx_din_imag }
 - Έξοδοι : { rx_dout_valid, rx_dout_byte, rx_dout_error }

⁸ Τη στιγμή που ένα σήμα ενός bit, διασχίζει το σύνορο μεταξύ δύο clock domain, ουσιαστικά «διαβάζεται» από τον καταχωρητή προέλευσης στο ένα ρολόι και «γράφεται» στον καταχωρητή προορισμού στο άλλο ρολόι. Λόγω της ανεξαρτησίας των ρολογιών, υπάρχει η πιθανότητα εμφάνισης του φαινομένου της μεταστάθειας στον καταχωρητή προορισμού. Τότε, ο καταχωρητής βρίσκεται σε μία ασαφή κατάσταση 'U', μεταξύ '1' και '0', δημιουργώντας ασάφεια στην τιμή εξόδου του.

⁹ Τα εσωτερικά σήματα των μονάδων δεν υπακούουν, απαραίτητα, σε αυτόν τον κανόνα, κυρίως σε περιπτώσεις αριθμητικών υπολογισμών όπου απαιτείται μεγάλη ενδιάμεση ακρίβεια. Το θέμα αυτό θα αναλυθεί διεξοδικά στο Κεφάλαιο 4.

¹⁰ Ως δυναμικό εύρος ενός τύπου αναπαράστασης, νοείται το διάστημα των αριθμών που μπορούν να αναπαρασταθούν με αυτόν.



Σχήμα 3.3: RXTXTOP - Chip I/O Pin Diagram

Από αυτούς οι

clk_s, clk_f, rst_f_n, rst_s_n

αποτελούν τις εισόδους των ρολογιών (clocks) και σημάτων επαναφοράς (resets) του συστήματος, οι

tx_din_byte, rx_din_real, rx_din_imag

αποτελούν τις εισόδους δεδομένων (data) του συστήματος, οι

tx_dout_real, tx_dout_imag, rx_dout_byte

αποτελούν τις εξόδους δεδομένων, ενώ οι

tx_din_valid, tx_din_ready, tx_dout_valid, rx_din_valid, rx_dout_valid

αντιστοιχούν σε σήματα ελέγχου της ροής δεδομένων, για τα οποία θα γίνει λεπτομερής αναφορά πιο κάτω σε αυτό το κεφάλαιο.

```
entity rxtxtop is
  generic(
    -- SHARED TELCO
    -- OFDM/FRAME GEN
    RXDIN_WIDTH : natural := 16; -- RX data in width
    TXDOUT_WIDTH : natural := 16; -- TX data out width
    ---- TELCO PARAMS
    SZ_CP : natural := 32; -- Cyclic prefix size (samples)
    LOG2_SZ_USYM : natural := 7; -- LOG2(SZ_USYM) SZ_USYM = Useful symbol length, also size of
    FFT transform and number of subcarriers
    SZ_FRAME : natural := 10; -- Frame size, including training symbols
    METRIC_VALID_THRES : natural := 60; -- Percentage of metric max value that defines if metric
    will be considered valid
    METRIC_SEARCH_SPAN : natural := 64; -- Metric Observation window size in samples (Z)
    TYP:0.5*SZ_USYM; MAX: 1*SZ_USYM=1024
    SOF_SAFE_MARGIN : natural := 8; -- Estimated start of frame offset, Optimal : (SZ_CP -
    SZ_CHAN) /2
    -- SCALING POST FFT
  );
end entity;
```



```

TX_POSTFFT_SHIFT_BITS : natural := 4; -- POSTFFT DAC SCALING (shift right bits)
RX_POSTFFT_SHIFT_BITS : natural := 5; -- POSTFFT RX SCALING (shift right bits)
-- SIMPLEX/HALF DUPLEX MODE
DUPLEX_MODE : natural := 0 -- 0: SIMPLEX TX | 1: SIMPLEX RX (Half DUPLEX not implemented
yet)
);
port(
  clk_s : in std_logic;
  rst_s_n : in std_logic;
  clk_f : in std_logic;
  rst_f_n : in std_logic;
----- T X -----
-- Avalon Streaming Data Sink Input Interface
tx_din_valid : in std_logic;
tx_din_byte : in std_logic_vector(7 downto 0);
tx_din_ready : out std_logic;
-- Avalon Streaming Source Output Interface
tx_dout_valid : out std_logic; --common [tx]
tx_dout_real : out std_logic_vector (TXDOUT_WIDTH - 1 downto 0);
tx_dout_imag : out std_logic_vector (TXDOUT_WIDTH - 1 downto 0);
tx_dout_error : out std_logic; --common [tx]
----- R X -----
-- Avalon Streaming Data Sink Input Interface
rx_din_real : in std_logic_vector (RXDIN_WIDTH - 1 downto 0);
rx_din_imag : in std_logic_vector (RXDIN_WIDTH - 1 downto 0);
rx_din_valid : in std_logic;
-- Avalon Streaming Source Output Interface
rx_dout_error : out std_logic; --common [rx]
rx_dout_valid : out std_logic; --common [rx]
rx_dout_byte : out std_logic_vector(7 downto 0)
);
end rxtxtop;

```

Ακολουθεί μια συνοπτική περιγραφή του κάθε ακροδέκτη :

Κοινοί (TX & RX) Ακροδέκτες

clk_s ▷ Πρόκειται για το κυρίως ρολόι του συστήματος. Τροφοδοτεί όλες τις μονάδες.

clk_f ▷ Το "γρήγορο" ρολόι, παράγεται απ' το κυρίως ρολόι του του συστήματος με χρήση PLL. Χρησιμοποιείται μόνο από το *Altera OFDM Megacore*

rst_s_n ▷ Το σήμα reset που αντιστοιχεί στο κυρίως ρολόι. Παρέχεται ως είσοδος σε όλες τις μονάδες. Είναι σήμα αρνητικής λογικής ('1' → ανενεργό, '0' → ενεργό)

rst_f_n ▷ Όπως και το rst_s_n είναι σήμα αρνητικής λογικής, αντιστοιχεί στο γρήγορο ρολόι του συστήματος και παρέχεται εσωτερικά ως είσοδος μόνο στο *Altera OFDM Megacore*

Ακροδέκτες Πομπού (TX)

tx_din_byte ▷ Είσοδος δεδομένων στον TX, εύρους 8 bit.

tx_din_valid ▷ Σήμα ελέγχου που σηματοδοτεί την εγκυρότητα (ή όχι) των δεδομένων εισόδου που βρίσκονται στη θύρα tx_din_byte.

tx_din_ready ▷ Αν και προθεματισμένο ως είσοδος, στην πραγματικότητα αποτελεί τη θύρα εξόδου του σήματος ελέγχου in_ready. Σηματοδοτεί την ευχέρεια (ή όχι) της καταβόθρας δεδομένων (data sink) να δεχτεί δεδομένα.

tx_dout_real / tx_dout_imag ▷ Δύο 16μπιτες έξοδοι δεδομένων. Αποτελούν τις (I,Q) συνιστώσες του προς μετάδοση OFDM σήματος. Η φυσιολογική εξέλιξη αυτών των εξόδων είναι το πέρασμα από τους DAC, η μίξη τους στο IF στάδιο, η ανύψωσή τους στα RF και η εκπομπή τους από την κεραία.

tx_dout_valid ▷ Σήμα ελέγχου που σηματοδοτεί την εγκυρότητα (ή όχι) των δεδομένων εξόδου που βρί-

σκονται στις θύρες `tx_dout_real` και `tx_dout_imag`.

tx_dout_error ▷ Ενδείκτης πιθανών σφαλμάτων κατά τη διαδικασία μετάδοσης των δεδομένων εξόδου.

Ακροδέκτες Δέκτη (RX)

rx_din_real / rx_din_imag ▷ Δύο 16μπιτες εισοδοι δεδομένων. Αποτελούν τις (I,Q) συνιστώσες του OFDM σήματος. Η προϊστορία τους έχει ως εξής : Λήψη από την κεραία του δεκτη, υποβιβασμός στα IF και διαχωρισμός σε (I,Q) συνιστώσες και, κατόπιν, ψηφιοποίηση στους ADC.

rx_din_valid ▷ Σήμα ελέγχου που σηματοδοτεί την εγκυρότητα (ή όχι) των δεδομένων εισόδου που βρίσκονται στις θύρες `rx_din_real` και `rx_din_imag`.

rx_dout_byte ▷ Έξοδος δεδομένων στον RX, εύρους 8 bit.

rx_dout_valid ▷ Σήμα ελέγχου που σηματοδοτεί την εγκυρότητα (ή όχι) των δεδομένων εξόδου που βρίσκονται στη θύρα `rx_dout_byte`.

rx_dout_error ▷ Ενδείκτης πιθανών σφαλμάτων κατά τη διαδικασία λήψης και ανακατασκευής του αρχικού μηνύματος πληροφορίας.

3.1.3 Παραμετροποίηση Συστήματος

Σύμφωνα με όσα αναφέρθηκαν στο κεφάλαιο 2 (§2.2.2), το σύστημα προσφέρει αρκετή ευελιξία σε σχέση με την παραμετροποίησή του.

Πρώτου επεξηγηθούν οι παράμετροι, κρίνεται απαραίτητο να γίνει μία διευκρίνιση: Κάθε φορά που γίνεται compile (σύνθεση-χωροθέτηση) ο κώδικας VHDL που αναπαριστά το κύκλωμα, δημιουργείται ένα "στιγμιότυπο" (instance) του κυκλώματος σε μορφή netlist. Ο προορισμός του είναι να κωδικοποιηθεί σε κατάλληλη μορφή και να μεταφερθεί πάνω σε ένα FPGA του είδους για το οποίο έχει συντεθεί. Εκεί μπορεί το κύκλωμα να "τρέξει" ή, πιο σωστά, να λειτουργήσει ως ψηφιακό κύκλωμα.

Παράμετροι

Στο κύκλωμα, παρέχεται ένα σύνολο παραμέτρων, που ορίζονται ως generic σταθερές στο top level component που απαρτίζει το σύστημα (όπως έχει παρουσιαστεί στην §3.1.2). Κάθε μία από αυτές μπορεί να αλλαχθεί εύκολα, ωστόσο αυτό είναι δυνατό μόνο σε διαφορετικά στιγμιότυπα του κυκλώματος.

Το σύνολο των παραμέτρων θα μπορούσε να χωριστεί εννοιολογικά σε δύο κατηγορίες :

- Παράμετροι που αφορούν στις τηλεπικοινωνιακές προδιαγραφές του συστήματος
- Παράμετροι που αφορούν στο σύστημα και την αρχιτεκτονική του.

Για τις περισσότερες τηλεπικοινωνιακές παραμέτρους έχει ήδη γίνει λόγος, στις §2.1.7 και 2.2.3, ωστόσο εισάγονται και κάποιες καινούριες που προκύπτουν από τον τρόπο που έχει υλοποιηθεί το κύκλωμα.

Άλλη μία κατηγοριοποίηση των παραμέτρων είναι και η επόμενη :

- Παράμετροι για τις οποίες δίνεται στο χρήστη δυνατότητα επιλογής τιμής (από ένα προκαθορισμένο σύνολο)
- Παράμετροι οι οποίες παραμένουν "κλειδωμένες", χωρίς ο τελικός χρήστης να έχει τη δυνατότητα τροποποίησης.

Μιλώντας χοντρικά, οι παράμετροι της πρώτης κατηγορίας αφορούν κυρίως στις τηλεπικοινωνιακές προδιαγραφές του συστήματος, ενώ οι παράμετροι της δεύτερης αφορούν σε συστημικές προδιαγραφές και αποτελούν βασικές παραμέτρους της αρχιτεκτονικής του συστήματος.

Το σύνολο των παραμέτρων του κυκλώματος αναγράφεται στον πίνακα 3.2, όπου σημειώνεται ο τύπος της παραμέτρου και η αντιστοιχία με γνωστά μεγέθη (τα μεγέθη που σημειώνονται με '*' ορίζονται για πρώτη φορά εδώ).

Όνομα Παραμέτρου	Τύπος	Τυπικό Όνομα	Περιγραφή
<i>LOG2_SZ_USYM</i>	<i>T</i>	$\log_2(N_o)$	μήκος του ωφέλιμου OFDM συμβόλου (λογάριθμος)
<i>SZ_CP</i>	<i>T</i>	N_{CP}	μήκος του cyclic prefix
<i>SZ_FRAME</i>	<i>T</i>	R	μέγεθος frame
<i>METRIC_VALID_THRES</i>	<i>T</i>	M_{rthres}	κατώφλι εγκυρότητας της μετρικής
<i>METRIC_SEARCH_SPAN</i>	<i>T</i>	M_{span}	εύρος αναζήτησης μετρικής
<i>SOF_SAFE_MARGIN</i>	<i>T</i>	S_{offset}	offset αρχής του frame
<i>TX_POSTFFT_SHIFT_BITS</i>	<i>T</i>	F_{TX} (*)	συντελεστής post fft κλίμακας για το tx path
<i>RX_POSTFFT_SHIFT_BITS</i>	<i>T</i>	F_{RX} (*)	συντελεστής post fft κλίμακας για το rx path
<i>RXDIN_WIDTH</i>	<i>S</i>	(*)	εύρος εξόδου του TX path
<i>DUPLEX_MODE</i>	<i>S</i>	(*)	τρόπος πολυπλεξίας του κυκλώματος

Πίνακας 3.2: Παράμετροι του Top-Level (RXTXTOP) • T: Τηλεπικοινωνιακή | S: Συστήματος

Επεξηγήσεις

Τηλεπικοινωνιακές παράμετροι

- F_{TX}, F_{RX} ▷ Υπάρχουν δύο μηχανισμοί κλιμάκωσης της εξόδου του IFFT/FFT που χρησιμοποιείται στον πομπό και το δέκτη, αντίστοιχα. Οι συντελεστές είναι δυνάμεις του 2, ώστε να είναι εύκολα υλοποιήσιμες στο υλικό με ολισθητές. Οι παράμετροι αυτές, ορίζουν τον αριθμό των ολισθήσεων προς τα δεξιά, υλοποιώντας το συντελεστή κλιμάκωσης $1/2^{F_{TX}}$ και $1/2^{F_{RX}}$ αντίστοιχα. Η σημασία τους σχετίζεται με την υλοποίηση του FFT/IFFT ζεύγους και θα αποσαφηνιστεί στην §3.2.4.
- S_{offset} ▷ Ορίζει το offset θεώρησης της αρχής του frame σε σχέση με το σημείο μεγιστοποίησης της μετρικής. Είναι παραγόμενο μέγεθος, καθώς η βέλτιστη τιμή του (όπως έχει εξηγηθεί στην §2.2.3) είναι: $S_{offset} = \frac{N_{CP} - T_h}{2}$.
- $\log_2(N_o)$ ▷ Σύνολο επιτρεπτών τιμών: $\log_2(N_o) \in \{6, 7, 8, 9, 10\}$. Η ανάλυση του κάτω ορίου γίνεται στην §4.7.1, ενώ το άνω όριο έχει τέθει ως προδιαγραφή του συστήματος.
- R ▷ Ισχύει $\max(R) = \frac{8192}{N_s} + 3$. Στο σύστημα, όπου $N_s = 160 \rightarrow \max(R) = 54$. Λεπτομερέστερη αναφορά γίνεται στην §4.7.

Συστημικές παράμετροι

- $TXDOUT_WIDTH$ ▷ Το εύρος εξόδου του TX path. Εξαρτάται από το εύρος του DAC στον οποίο θα συνδεθεί το κύκλωμα.
- $RXDIN_WIDTH$ ▷ Το εύρος της εισόδου του RX path. Εξαρτάται από το εύρος του ADC στον οποίο θα συνδεθεί το κύκλωμα.
- $DUPLEX_MODE$ ▷ Ρυθμίζει τον τρόπο πολυπλεξίας του κυκλώματος. Η κωδικοποίηση του ερμηνεύεται στον ακόλουθο πίνακα (3.3):

DUPLEX_MODE	Λειτουργία
0	SIMPLEX TX
1	SIMPLEX RX
2	HALF-DUPLEX (μελλοντική επέκταση)

Πίνακας 3.3: Κωδικοποίηση παραμέτρου DUPLEX_MODE

Asset

Εκτός από τις παραμέτρους, υπάρχουν και άλλα στοιχεία που θεωρούνται σταθερά για ένα στιγμιότυπο του κυκλώματος, τα λεγόμενα asset. Τα τελευταία αποθηκεύονται είτε σε δομές ROM είτε ως κυκλωματικές σταθερές και χρησιμοποιούνται από διάφορα μέρη στο σύστημα. Κατασκευάζονται με τη βοήθεια του περιβάλλοντος τεχνικής υπολογιστικής *MATLAB* και είναι εύκολο να αλλαχθούν, στο βαθμό που αυτό έχει νόημα.

Τα asset είναι:

- *Training symbols συγχρονισμού* ▷ Είναι 2 στατικά, διαμορφωμένα κατά QAM-4, ωφέλιμα OFDM σύμβολα που εισάγονται στην αρχή κάθε frame. Χρησιμοποιούνται από τον αλγόριθμο Schmidl & Cox κατά το συγχρονισμό στο χρόνο.
- *Training symbol εκτίμησης καναλιού* ▷ Είναι 1 στατικό, διαμορφωμένο κατά QAM-4, ωφέλιμο OFDM σύμβολο που εισάγεται μετά τα training συγχρονισμού σε κάθε frame. Χρησιμοποιείται από τον αλγόριθμο (LS) εκτίμησης καναλιού.
- *Constellations* ▷ Οι αστερισμοί διαμόρφωσης κατά QAM που υποστηρίζει το σύστημα (QAM 4|16|64).
- *Modulation pattern* ▷ Το μοτίβο διαμόρφωσης ανα subcarrier.

Για την παραγωγή των asset έχουν υλοποιηθεί παραμετροποιήσιμα, βοηθητικά script του περιβάλλοντος *MATLAB*. Η έξοδος αυτών των script δεν είναι η τελική μορφή των asset αλλά μία ενδιάμεση, ώστε να διευκολυνθεί η αποσφαλμάτωση και η διαχείριση τους.

Επιπλέον κάποια asset πρέπει να αντιγραφούν σε διαφορετικά σημεία στο κύκλωμα, συχνά με διαφορετική κωδικοποίηση (π.χ. ο αστερισμός στους constellation mapper και demapper). Κατ' επέκταση έχει υλοποιηθεί και μία ομάδα από script *MATLAB* που διανέμουν τα assets στις κατάλληλες τοποθεσίες. Η διαδικασία αυτή καλείται *asset dispatching* και θα πρέπει να εκτελείται κάθε φορά που αλλάζει κάποιο από τα asset.

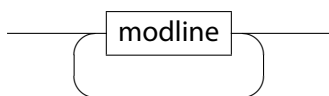
Μοτίβο Ψηφιακής Διαμόρφωσης (*mod_pattern*)

Είναι δυνατό να καθοριστεί το μοτίβο διαμορφώσεων (*modulation pattern*), δηλαδή της ακολουθίας διαμόρφωσης για κάθε subcarrier. Η ακολουθία επαναλαμβάνεται σε κάθε OFDM σύμβολο. Για τον ορισμό του *modulation pattern* καθορίζεται μία συμπιεσμένη μορφή αναπαράστασής του, την οποία εισάγει ο σχεδιαστής στο κύκλωμα. Η δομή αυτή είναι εξαιρετικά ευέλικτη, καθώς επιτρέπει ακολουθίες διαμορφώσεων διαδοχικών συμβόλων, όπως {4,16,16,64,16,4...} ή {4,4,4,4...} κλπ., σύμφωνα με τον παρακάτω κανόνα :

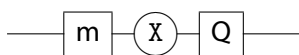
$$\begin{aligned}
 \text{MOD_PATTERN} &:: n \times \text{MOD}(\text{line}) \\
 \text{MOD}(\text{line}) &:: m \times Q \\
 Q &:: 4|16|64 \\
 m, n &\in \mathbb{N}
 \end{aligned}$$

ή, πιο παραστατικά :

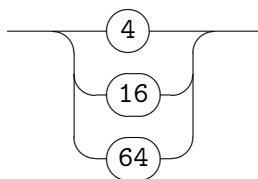
modpattern



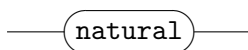
modline



Q



m



Το μοτίβο διαμόρφωσης είναι από κοινού γνωστό τόσο στον πομπό όσο και στο δέκτη και κάθε φορά, ο πομπός και ο δέκτης έχουν το ίδιο "στιγμιότυπο" (instance). Μια μελλοντική επέκταση θα ήταν η δημιουργία ενός προσαρμοστικού (adaptive) συστήματος διαμόρφωσης που θα λαμβάνει υπόψη τις τηλεπικοινωνιακές συγκυρίες στο κανάλι και αυτόματα θα προσαρμόζει το είδος της διαμόρφωσης QAM.

Αστερισμοί Διαμόρφωσης (*constellations*)

Οι τύποι διαμόρφωσης των subcarrier που υποστηρίζονται από το σύστημα είναι QAM-4, QAM-16, QAM-64. Η κάθε διαμόρφωση αντιστοιχεί σε έναν αστερισμό, δηλαδή σε ένα σύνολο μιγαδικών συμβόλων. Η μορφή των αστερισμών, όπως έχει αναφερθεί στην §2.2.1, είναι τετραγωνική. Επιπλέον η μέση ισχύς όλων των αστερισμών έχει κανονικοποιηθεί στη "μονάδα". Με αυτόν τον τρόπο γίνεται πιο αποδοτική εκμετάλλευση της διαθέσιμης ισχύος στους μικρούς αστερισμούς (QAM-4, QAM-16) και επιτυγχάνεται μεγαλύτερη ανοχή στο θόρυβο.

Ωστόσο, οι αριθμητικές τιμές που έχουν αποδοθεί στα "σημεία" των αστερισμών διαμόρφωσης θα πρέπει να αναπαρίστανται με το τύπο αριθμητικής του κυκλώματος, τον int16. Υπενθυμίζεται ότι το δυναμικό εύρος του int16 είναι $[-32768, 32767]$. Επιπλέον, υπάρχει η αναγκαιότητα ρύθμισης της «μοναδιαίας» ισχύος του σήματος που εισέρχεται στον IFFT.¹¹ Για την κάλυψη αυτών των αναγκών, εισάγεται (ως ποσοστό του $max(int16) = 32767$) ο συντελεστής κλιμάκωσης του αστερισμού (*CONSTEL_SCALE_INT16MAX_PRCNT*).

Σημειώνεται ακόμα πως της ίδιας κανονικοποίησης και κλιμάκωσης χρήζουν και οι βοηθητικοί αστερισμοί διαμόρφωσης των training symbol, έτσι ώστε να μην υπάρχει αλλαγή στη μέση ισχύ κατά τη μετάδοσή τους. Αυτή είναι μία διαδικασία που επιτελείται στο σχετικό script MATLAB που παράγει τα training symbol.

Θα πρέπει να αναφερθεί, τέλος, ότι τα training σύμβολα αποθηκεύονται σε ROM τόσο στον TX όσο και στον RX.

¹¹Θα γίνει εκτενής αναφορά στο θέμα προ- και μετά- FFT κλιμάκωσης στις §§1.2.4, 1.2.5

Διανομή παραμέτρων (*parameter dispatching*)

Ανακεφαλαιώνοντας, στο κύκλωμα υπάρχουν κάποια σταθερά στοιχεία για κάθε στιγμιότυπό του. Πρόκειται για τις παραμέτρους του top-level component και για τα asset. Οι παράμετροι και τα asset προσδιορίζουν, ως επί το πλείστον, την τηλεπικοινωνιακή συμπεριφορά του κυκλώματος. Συνεπώς είναι επιθυμητό να μπορούν να αλλάζουν κατά βούληση κατά τη διαδικασία της σχεδίασης του συστήματος. Για την υποστήριξη αυτής της διαδικασίας έχει υλοποιηθεί ένας μηχανισμός ολοκληρωμένης διαχείρισης των παραμετροποιήσιμων μεγεθών του κυκλώματος (*asset generation, asset-parameter dispatching*) σε περιβάλλον MATLAB. Οι αλλαγές γίνονται σε ένα σημείο και διαχέονται αυτόματα σε όλα τα μέρη του κυκλώματος που χρειάζεται. Με αυτόν τον τρόπο εξασφαλίζεται η συνέπεια μεταξύ των διαφορετικών μεγεθών και επιταχύνεται η διαδικασία της παραμετροποίησης.

Ενας άλλος λόγος της ύπαρξης του μηχανισμού αυτού είναι ότι τα asset και οι παράμετροι χρησιμοποιούνται τόσο από το κύκλωμα, όσο και από το μοντέλο του κυκλώματος στο MATLAB.¹² Σημειώνεται πως κάποιες παραγόμενες παράμετροι υπολογίζονται αυτόματα από άλλα, αρχικά μεγέθη. Το σύνολο των τελικών τηλεπικοινωνιακών παραμέτρων, όπως αυτές είναι διαθέσιμες από το μηχανισμό του *asset-parameter dispatching*, συνοψίζεται στον πίνακα 3.4.

Όνομα Παραμέτρου	Τυπικό Όνομα	Περιγραφή
<i>LOG2_SZ_USYM</i>	$\log_2(N_o)$	μήκος του ωφέλιμου OFDM συμβόλου (λογάριθμος)
<i>SZ_CP</i>	N_{CP}	μήκος του cyclic prefix
<i>SZ_FRAME</i>	R	μέγεθος frame
<i>METRIC_VALID_THRES</i>	M_{rthres}	κατώφλι εγκυρότητας της μετρικής
<i>METRIC_SEARCH_SPAN</i>	M_{span}	εύρος αναζήτησης μετρικής
<i>TX_POSTFFT_SHIFT_BITS</i>	$F_{TX} (*)$	συντελεστής post fft κλίμακας για το tx path
<i>RX_POSTFFT_SHIFT_BITS</i>	$F_{RX} (*)$	συντελεστής post fft κλίμακας για το rx path
<i>CONSTEL_SCALE_INT16MAX_PRCNT</i>	-	συντελεστής κλιμάκωσης αστερισμού

Πίνακας 3.4: Διανεμόμενες (dispatched) Τ/Π παράμετροι

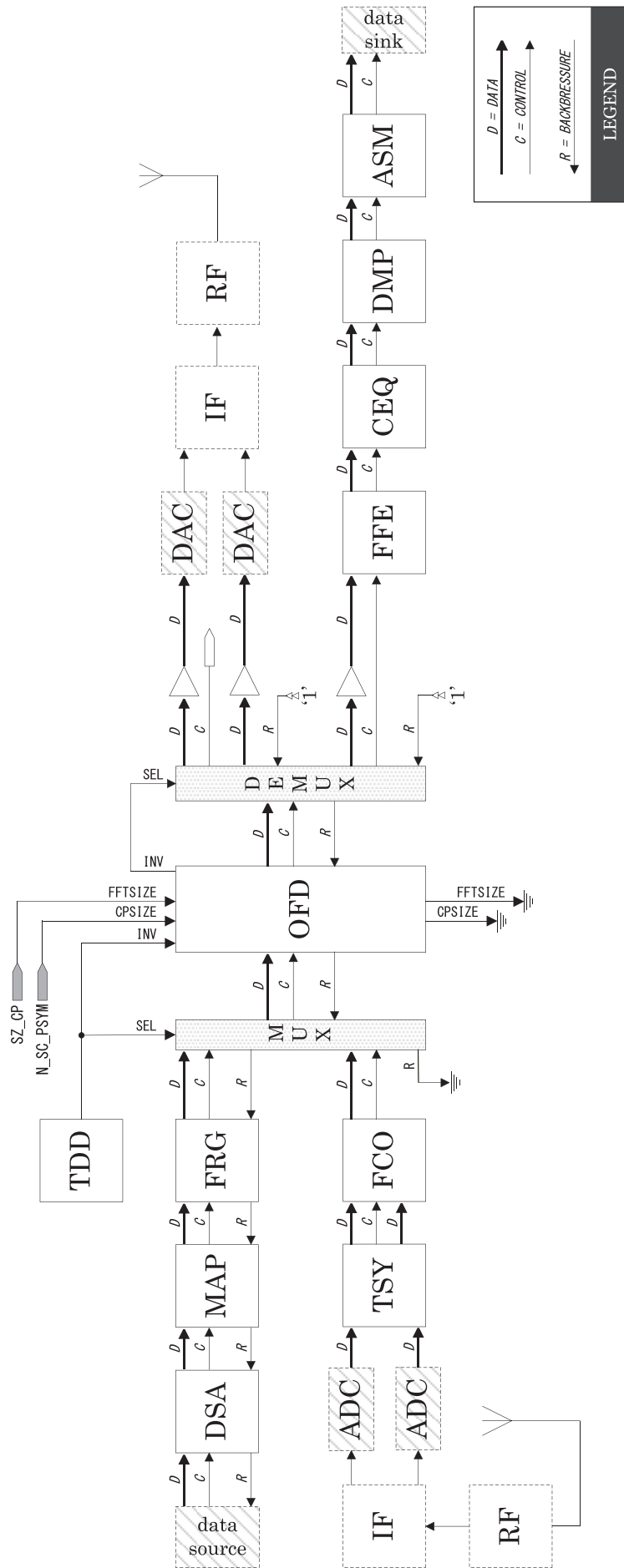
3.2 Πλήρης Αλυσίδα Συστήματος Πομποδέκτη

Στο διάγραμμα 3.4 εμφανίζεται, σε συνεπτυγμένη μορφή, ολόκληρος ο διάδρομος δεδομένων (εφεξής *datapath*) και περιλαμβάνει τόσο το διάδρομο πομπού (*TX-PATH*) όσο και του δέκτη (*RX-PATH*). Απεικονίζεται, έτσι, το σύστημα στην ολότητά του ακριβώς όπως έχει υλοποιηθεί. Τα ορθογώνια "κουτιά" αποτελούν τις βασικές υπομονάδες (modules) του συστήματος και σε αυτό το κεφάλαιο θα παραμείνουν ως έχουν, δηλαδή σα *μαύρα κουτιά*, μιας και αντικείμενο του κεφαλαίου είναι η συστημική θεώρηση του πομποδέκτη και όχι η αναλυτική περιγραφή των υπομονάδων του, η οποία θα ακολουθήσει στο Κεφάλαιο 4. Ο Πίνακας 3.5 παρουσιάζει την αντιστοιχία μεταξύ των ονομάτων των modules και των συντομογραφιών που χρησιμοποιούνται στα διαγράμματα.

Σε αυτό το σημείο θα πρέπει να τονιστεί ότι το module *Altera OFDM Core* (OFD στο διάγραμμα) χρησιμοποιείται τόσο στη λειτουργία πομπού όσο και στη λειτουργία δέκτη.

Σκοπός του παρόντος κεφαλαίου, λοιπόν, είναι η ανάδειξη του τρόπου διασύνδεσης των λειτουργικών υπομονάδων μεταξύ τους, η περιγραφή της διεπαφής (*interface*) που ορίζει ένα "πρωτόκολλο" επικοινωνίας μεταξύ των υπομονάδων και η διαλεύκανση της πορείας των δεδομένων, μέσα στο σύστημα, μέσα από τις περιγραφές των *TX-PATH* και *RX-PATH*. Ως πιο σωστή αφετηρία, κρίνεται η περιγραφή του *interface* που χρησιμοποιείται για τις "συνδιαλλαγές" των modules, τόσο όσον αφορά στα δεδομένα καθεαυτά, όσο και στα σήματα ελέγχου.

¹²Περισσότερα για το μοντέλο MATLAB θα αναφερθούν στο Κεφάλαιο 5



Σχήμα 3.4: Συνοπτικό RX/TX TOP Datapath

Σύντμηση	Πλήρες Όνομα
DSA	Dissassembler
MAP	Symbol Mapper
FRG	Frame Generator
TDD	Ελεγκτής Τρόπου Μετάδοσης
OFD	Altera OFDM Core
TSY	Time Synchronizer
FSO	Frequency Compensator
FFE	Fine Frequency Estimator
CEQ	Channel Estimator (Equaliser)
DMP	Symbol Demapper
ASM	Assembler

Πίνακας 3.5: Ακρωνύμια υπομονάδων (modules)

3.2.1 Διεπαφή Επικοινωνίας Υπομονάδων

Η αποδοχή κάποιου συγκεκριμένου interface και ο ορισμός, έτσι, ενός πρωτοκόλλου επικοινωνίας ανάμεσα στις υπομονάδες που απαρτίζουν ένα σύστημα είναι ένα βασικό στάδιο κατά τη διαδικασία της συστημικής σχεδίασης. Πράγματι, η επιλογή του κατάλληλου interface αφενός διευκολύνει τη λειτουργική σχεδίαση και αποσφαλμάτωση και αφετέρου βάζει μία τάξη στα σήματα ελέγχου και στον τρόπο μεταγωγής των δεδομένων. Εξάλλου, μια βασική απαίτηση των σύγχρονων συστημάτων είναι η ευκολία αναβάθμισης και τροποποίησης των επιμέρους λειτουργιών τους, αν αυτό κριθεί σκόπιμο. Ένας πολύτιμος σύμμαχος είναι η *διαμορφωσιμότητα* (modularity) του συστήματος. Για παράδειγμα, αν κριθεί σκόπιμη η επανασχεδίαση ενός module -λόγω π.χ. εύρεσης ενός καλύτερου αλγορίθμου- ο σχεδιαστής μπορεί να το αναπτύξει εκ νέου σε επίπεδο λειτουργικής μονάδας (unit) και κατόπιν, εφόσον ακολουθήσει τις επιταγές του interface του συστήματος, να το "κρεμάσει" στην αλυσίδα χωρίς περαιτέρω έγνοιες για τη συστημική συμπεριφορά της νέας υπομονάδας.

Στην παρούσα υλοποίηση επιλέχθηκε το *Altera Avalon Streaming Interface* ως η βασική διεπαφή των υπομονάδων του κυρίως κορμού του συστήματος, καθώς το εν λόγω interface¹³ είναι σχεδιασμένο για επικοινωνία υποσυστημάτων που διαχειρίζονται δεδομένα μονής κατεύθυνσης (unidirectional), υψηλού bandwidth και μικρής καθυστέρησης, όπως ακριβώς είναι η φύση των δεδομένων σε εφαρμογές DSP.¹⁴

Προδιαγραφή του Altera Avalon ST - Streaming Interface

Πριν γίνει οποιαδήποτε απόπειρα παρουσίασης του Avalon Interface δίνεται (Πίνακας 3.6) ένα μικρό γλωσσάρι όρων και εννοιών που σχετίζονται με αυτό.

Συστατικά (Components)	Εναλλακτική ονομασία για τις λειτουργικές βαθμίδες ή υποσυστήματα ή modules
Πάνω-ρεύμα - Upstream	Δεδομένου ενός component, upstream ονομάζεται το σύνολο των component που βρίσκονται πιο πριν από αυτό -στην αλυσίδα-, κατά την κατεύθυνση ροής τη πληροφορίας.
Κάτω-ρεύμα - Downstream	Δεδομένου ενός component, downstream ονομάζεται το σύνολο των component που βρίσκονται μετά από αυτό -στην αλυσίδα-, κατά την κατεύθυνση ροής τη πληροφορίας.

¹³Προδιαγράφεται στο [Ava]

¹⁴Σε αυτό το σημείο θα πρέπει να διευκρινιστεί ότι το interface είναι μια προδιαγραφή, μια συλλογή τύπων και συμβάσεων και όχι υλοποίηση. Η υλοποίηση των προδιαγραφών είναι εργασία που επιτελεί ο σχεδιαστής του συστήματος. Αυτός καθορίζει, επίσης, το ποσοστό κάλυψης των προδιαγραφών που ορίζει το interface, επιλέγοντας (ή όχι) την υλοποίηση μιας συγκεκριμένης προδιαγραφής, κατά το δοκούν.

Διεπαφές Source & Sink - Συνδέσεις (Connections)	Όταν συνδέονται δυο συστατικά, τα δεδομένα ρέουν από τη διεπαφή πηγής (source) προς τη διεπαφή καταβόθρας (sink). Ο συνδυασμός ενός source και ενός sink interface ονομάζεται σύνδεση (connection).
Όπισθεν Πέδηση (Backpressure)	Μηχανισμός με τον οποίο ο sink ειδοποιεί το source να σταματήσει την αποστολή δεδομένων. Ο μηχανισμός χρησιμοποιείται, τυπικά, όταν οι FIFO του sink είναι γεμάτοι ή όταν υπάρχει συμφόρηση στη θύρα εξόδου του.
Έμπροσθεν Πέδηση (Frontpressure)	Μηχανισμός με τον οποίο ο source ειδοποιεί το sink ότι στη θύρα εισόδου του δεν υπάρχουν έγκυρα δεδομένα και ως εκ τούτου οφείλει να τα αγνοήσει. Ο μηχανισμός χρησιμοποιείται, τυπικά, όταν ο source δεν έχει ολοκληρώσει ακόμη την επεξεργασία των δεδομένων ή όταν έχει προκύψει κάποιο πρόωρο έγκυρο αποτέλεσμα, το οποίο πρέπει να καθυστερήσει για λόγους συγχρονισμού.
Μεταφορές (Transfers)	Λειτουργία που καταλήγει στη διάδοση πληροφορίας και σημάτων ελέγχου από μια διεπαφή source σε μια διεπαφή sink.
Κύκλοι Δεδομένων (Data cycles)	Κύκλοι κατά τους οποίους γίνεται μεταφορά δεδομένων και σημάτων ελέγχου από μια διεπαφή source σε μια διεπαφή sink.
Αδρανείς Κύκλοι (Idle Cycles)	Κύκλοι κατά τους οποίους δεν επιτελούνται μεταφορές είτε λόγω frontpressure, είτε λόγω backpressure.
Κύκλοι Ετοιμότητας (Ready Cycles)	Κύκλοι κατά τους οποίους μια διεπαφή sink δύναται να αποδεχτεί μία μεταφορά.
Σύμβολο (Isymbol - Iunit)	Το βασικό δομικό στοιχείο πληροφορίας που μεταφέρεται ανά θύρα. Ένα ή περισσότερα σύμβολα συναποτελούν μία μονάδα πληροφορίας (iunit) που μεταφέρεται ανά κύκλο δεδομένων.
Πακέτο (IPacket)	Πρόκειται για μια συσσωμάτωση δεδομένων και σημάτων ελέγχου που μεταδίδονται από κοινού. Η μορφή του πακέτου καθορίζεται από τη φύση της εφαρμογής και όχι από την προδιαγραφή της διεπαφής.

Πίνακας 3.6: Γλωσσάρι για το Avalon ST Interface

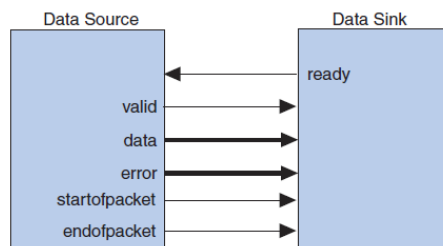
Ακολουθεί μια περιγραφή (Πίνακας 3.7) των βασικών σημάτων (ή καλύτερα των ρόλων που αυτά διαδραματίζουν) στη λειτουργική "θέσπιση" του Avalon Interface.¹⁵

Ρόλος	Κατεύθυνση	Περιγραφή
<i>data</i>	<i>source</i> → <i>sink</i>	Φέρει τον κύριο όγκο των δεδομένων. Τα περιεχόμενα και η δομή καθορίζονται από το σχεδιαστή.
<i>error</i>	<i>source</i> → <i>sink</i>	Μια μάσκα bit που σηματοδοτεί λάθη που εμφανίζονται στα δεδομένα που μεταφέρονται στον τρέχων κύκλο. Κάθε bit προδιαγράφει και διαφορετικό τύπο λάθους ανά εφαρμογή.
<i>valid</i>	<i>source</i> → <i>sink</i>	Σηματοδοτείται από το source προκειμένου να πιστοποιηθεί η εγκυρότητα των υπολοίπων σημάτων με κατεύθυνση από το source στο sink. Σε κύκλους ready όταν το σήμα valid είναι ενεργό, ο διάδρομος δεδομένων (data bus) αλλά και τα υπόλοιπα σήματα εισόδου δειγματοληπτούνται από το sink. Σε αντίθετη περίπτωση αγνοούνται. Χρησιμοποιείται για την υλοποίηση του μηχανισμού <i>frontpressure</i> .
<i>ready</i>	<i>sink</i> → <i>source</i>	Η ενεργοποίησή του υποδηλώνει ότι ο sink δύναται να δεχτεί δεδομένα. Ο sink ενεργοποιεί το σήμα ready στον κύκλο <n> για να ορίσει τον κύκλο <n + readyLatency> ως κύκλο ready, κατά τη διάρκεια του οποίου ο source μπορεί να ενεργοποιήσει το σήμα valid και να εκκινήσει, έτσι, η λειτουργία της μεταφοράς. Χρησιμοποιείται για την υλοποίηση του μηχανισμού <i>backpressure</i> .

¹⁵Όλα τα σήματα ενεργοποιούνται στο λογικό '1'.

<i>startofpacket (sop)</i>	source → sink	Ενεργοποιείται από το source για να σηματοδοτήσει την αρχή ενός πακέτου (<i>packet</i>). Λαμβάνεται υπόψη μόνο όταν το σήμα <i>valid</i> είναι ενεργοποιημένο.
<i>endofpacket (eop)</i>	source → sink	Ενεργοποιείται από το source για να σηματοδοτήσει το τέλος ενός πακέτου (<i>packet</i>). Λαμβάνεται υπόψη μόνο όταν το σήμα <i>valid</i> είναι ενεργοποιημένο.

Πίνακας 3.7: Ρόλοι σημάτων στο Avalon ST Interface



Σχήμα 3.5: Τυπικά σήματα του Avalon ST Interface

Συμπληρωματικά

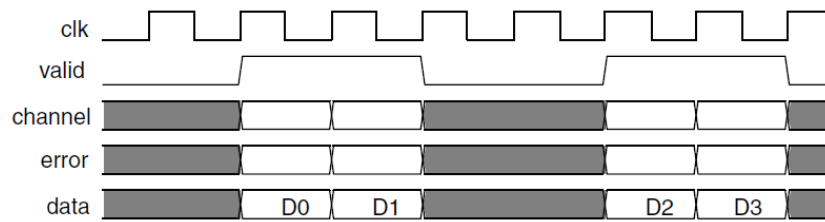
- Όλες οι μεταφορές σε μια σύνδεση (*connection*) λαμβάνουν χώρα σύγχρονα με την ανερχόμενη ακμή (*rising edge*) του σχετικού ρολογιού. Όλες οι έξοδοι μίας διεπαφής *source* που συνδέονται με εισόδους μίας διεπαφής *sink*, συμπεριλαμβανομένου του διαδρόμου δεδομένων και της δομής *error*, πρέπει να καταχωρούνται (*register*) στην ανερχόμενη ακμή του ρολογιού. Οι εισοδοι στο *sink* δε χρειάζονται καταχώρηση.
- Δεν είναι απαραίτητο να περιλαμβάνεται στα *modules* η θύρα ελέγχου ρολογιού (*clock_enable*), καθώς το σύστημα σηματοδότησης του Avalon ST είναι αρκετό ώστε να καθοριστούν, επακριβώς, οι κύκλοι κατά τους οποίους ένα *module* οφείλει (ή όχι) να είναι ενεργοποιημένο. Εσωτερικά, βέβαια, τα *module* μπορούν να έχουν μια τέτοια θύρα για την εσωτερική τους λογική, θα πρέπει όμως να ληφθεί μέριμνα ώστε ο συγχρονισμός των σημάτων ελέγχου να διασφαλίζει τις αρχές του πρωτοκόλλου.
- *readyLatency* :: Έγκυρες τιμές [0-8] :: Καθορίζει τη χρονική σχέση της ενεργοποίησης και απενεργοποίησης του σήματος *ready* και των κύκλων που θεωρούνται έγκυροι για μεταφορά δεδομένων (*ready cycles*), ξεχωριστά για κάθε *interface*.

Μεταφορά δεδομένων απουσία όπισθεν πέδησης

Η μεταφορά δεδομένων χωρίς την παρουσία *backpressure* αποτελεί την πιο βασική λειτουργία μεταφοράς του Avalon interface. Σε κάθε δεδομένο κύκλο ρολογιού, η διεπαφή *source* οδηγεί τα δεδομένα και το προαιρετικό σήμα λάθους, *error*, και ενεργοποιεί το σήμα *valid* σηματοδοτώντας, έτσι, την εγκυρότητα των δεδομένων. Η διεπαφή *sink* δειγματοληπτεί αυτά τα σήματα στην ανερχόμενη ακμή του ρολογιού, όταν το σήμα *valid* είναι ενεργό. Το σχήμα 3.6 απεικονίζει μία τυπική περίπτωση.

Έμπροσθεν/Οπίσθια Πέδηση - *Front/Back Pressure*

Η έννοια *frontpressure* σχετίζεται με την πέδηση (*stall*) που προέρχεται από το *upstream*. Στον πομπό δεν προβλέπεται μηχανισμός *frontpressure*, κατά την κανονική λειτουργία. Αυτό οφείλεται στο γεγονός ότι ο



Σχήμα 3.6: Παράδειγμα μεταφοράς δεδομένων χωρίς backpressure

DAC ορίζει το bitrate στην έξοδο του κυκλώματος, και μάλιστα, το bitrate αυτό είναι σταθερό. Ο DAC, δηλαδή, ορίζει το deadline για το datapath του πομπού. Το upstream, λοιπόν, δεν επιβάλλει frontpressure στον DAC, ειδάλλως θα παρουσιαστούν φουσαλλίδες (bubbles) και ενδεχομένως θα υπάρξει απώλεια δεδομένων (data loss). Στο δέκτη ο μηχανισμός αυτός είναι παρών. Όπως αναφέρθηκε, υλοποιείται με τη βοήθεια του σήματος *valid*.

Η έννοια backpressure σχετίζεται με την πέδηση (stall) που προέρχεται από το downstream. Ενώ στον πομπό ο μηχανισμός backpressure υφίσταται, στο δέκτη λόγω της παρουσίας του ADC, αυτό δε συμβαίνει. Συγκεκριμένα, στο δέκτη θεωρείται ότι κάθε module είναι διαρκώς έτοιμο να δεχτεί δεδομένα, το οποίο μεταφράζεται στην απουσία υλοποίησης του interface σήματος *ready*. Αυτό προκύπτει αφού, τώρα, ο ADC ορίζει το deadline του datapath του δέκτη. Αντίθετα στον πομπό το σήμα *ready* έχει υλοποιηθεί κανονικά και μάλιστα, όπως εξηγείται στην επόμενη παράγραφο, έχει επιλεγεί ως σχεδιαστική τιμή της παραμέτρου *readyLatency* το 0.

Στον πίνακα 3.8 περιγράφεται η άσκηση (ή μη) των μηχανισμών front/backpressure ανά module, τόσο για τον πομπό όσο και για το δέκτη.

Module	Σύστημα	F.P.	B.P.
Disassembler	TX	X	✓
Symbol Mapper	TX	X	X
Frame Generator	TX	X	✓
Altera OFDM • Modulator - IFFT	TX	X	✓
DAC	TX	X	X
ADC	RX	X	X
Time Synchronizer	RX	✓	X
Frequency Compensator	RX	X	X
Altera OFDM • Demodulator - FFT	RX	✓	X
"Fine" Frequency Estimator	RX	✓	X
Channel Estimator	RX	✓	X
Symbol Demapper	RX	X	X
Assembler	RX	✓	X

Πίνακας 3.8: Modules και Front/Back Pressure

Μεταφορά δεδομένων παρουσία όπισθεν πέδησης - *Ready Latency*

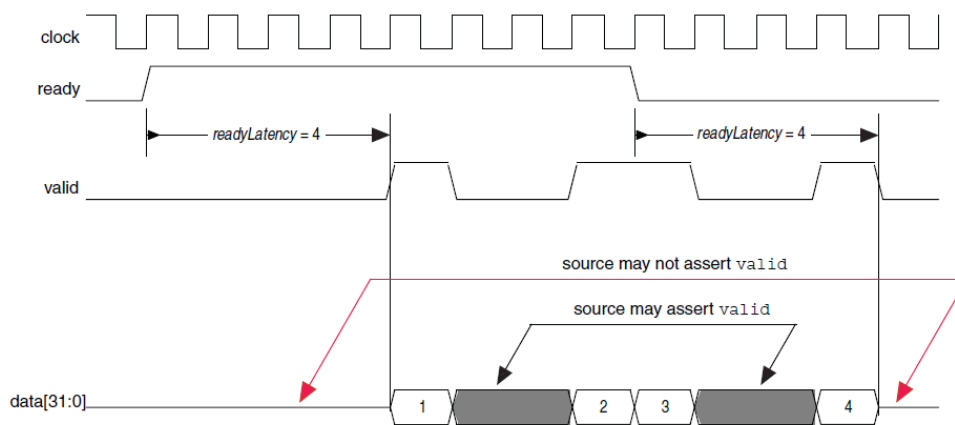
Κατά τη μεταφορά δεδομένων παρουσία όπισθεν πέδησης (backpressure), ο sink δύναται να ενημερώσει το source για τη δυνατότητά του να δεχτεί δεδομένα.¹⁶ Όπως προαναφέρθηκε, η παράμετρος *readyLatency* υποδηλώνει τον αριθμό των κύκλων ρολογιού, από τη στιγμή που το σήμα *ready* ενεργοποιείται έως ότου

¹⁶Υλοποιείται, δηλαδή, ένας μηχανισμός ελέγχου της ροής δεδομένων (data flow control)

υπάρξουν έγκυρα δεδομένα για οδήγηση. Ανάλογα με την τιμή της παραμέτρου *readyLatency* ορίζονται δύο τρόποι μεταφοράς παρουσία *backpressure*:

- *readyLatency* = 0
- *readyLatency* > 0

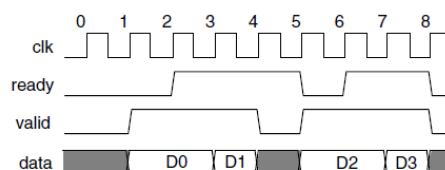
Όταν *readyLatency* ≥ 1 , ο sink ενεργοποιεί το σήμα *ready* πριν από την έλευση του καθαυτού *ready cycle*. Ο source μπορεί να ανταποκριθεί στη διάρκεια του κατάλληλου κυκλου οδηγώντας τα δεδομένα και ενεργοποιώντας το σήμα *valid*. Δεν έχει, ωστόσο, την ελευθερία να ενεργοποιήσει το *valid* κατά τη διάρκεια ενός μη *ready cycle*. Το σχήμα 3.7 παρουσιάζει ένα παράδειγμα όπου *readyLatency* = 4.



Σχήμα 3.7: Παράδειγμα μεταφοράς παρουσία *backpressure* (*readyLatency* = 4)

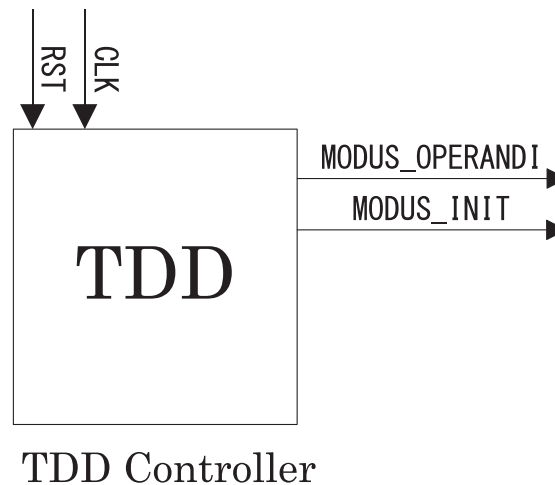
Όταν το *readyLatency* είναι ίσο με το 0, η μεταφορά δεδομένων λαμβάνει χώρα μόνο όταν τα σήματα *ready* και *valid* ενεργοποιούνται στον ίδιο κύκλο, ο οποίος είναι, φυσικά, ένας *ready cycle*. Σε αυτό τον τρόπο λειτουργίας, ο source δε λαμβάνει το σήμα *ready* από το sink πριν να έχει έγκυρα δεδομένα προς αποστολή. Ο source οδηγεί τα δεδομένα, ενεργοποιεί το σήμα *valid* οποτεδήποτε μπορεί και περιμένει τη ληψη των δεδομένων από τον sink, που σηματοδοτείται με την ενεργοποίηση του σήματος *ready*. Ο source μπορεί να αλλάξει τα δεδομένα που παρέχει ανα πάσα στιγμή. Ο sink μόνο τότε συλλαμβάνει τα δεδομένα εισόδου από το source, όταν και το σήμα *ready* και το σήμα *valid* είναι ενεργοποιημένα.

Στο διάγραμμα 3.8 παρουσιάζεται μία μεταφορά παρουσία *backpressure* και με *readyLatency* = 0. Ο source παρέχει δεδομένα και ενεργοποιεί το σήμα *valid* στον κύκλο 1, παρόλο που ο sink δεν είναι σε κατάσταση *ready* (βρίσκεται δηλαδή σε αδρανή -idle- κύκλο. Ο source περιμένει μέχρι τον κύκλο 2, όπου ο sink όντως ενεργοποιεί το σήμα *ready*, μέχρι να περάσει στον επόμενο κύκλο μεταφοράς δεδομένων. Στον κύκλο 3, ο source οδηγεί δεδομένα και ο sink είναι έτοιμος να τα δεχτεί, επομένως η μεταφορά λαμβάνει χώρα αμέσως, χωρίς κάποια καθυστέρηση. Στον κύκλο 4, ο sink έχει ενεργοποιημένο το σήμα *ready*, αλλά ο source δεν έχει έγκυρα δεδομένα προς μεταφορά (*valid* = 0).



Σχήμα 3.8: Παράδειγμα μεταφοράς παρουσία *backpressure* (*readyLatency* = 0)

3.2.2 Ελεγκτής Τρόπου Μετάδοσης



Σχήμα 3.9: Διάγραμμα Ακροδεκτών TDD Controller

Όπως έχει αναφερθεί το σύστημα μπορεί να λειτουργήσει με 2 τρόπους μετάδοσης: α) Simplex και β) Half Duplex με πολυπλεξία διαίρεσης χρόνου (TDD). Το module που καθορίζει τον τρόπο λειτουργίας του συστήματος είναι ο *tdd_controller*. Στο σχήμα 3.9 παρουσιάζεται η top-level περιγραφή του και ακολουθεί μια συνοπτική περιγραφή του κάθε ακροδέκτη.

```
entity tdd_ctrl is
  generic(
    DUPLEX_MODE: natural := 0 -- 0: SIMPLEX TX / 1: SIMPLEX RX (Half DUPLEX not implemented yet)
  );
  port(
    rst          : in std_logic;
    clk          : in std_logic;
    modus_operandi : out std_logic;
    modus_init    : out std_logic
  );
end tdd_ctrl;
```

Ακροδέκτες Εισόδου

clk ▷ Το βασικό ρολόι του συστήματος.

rst ▷ Το βασικό σήμα επαναφοράς του συστήματος.

Ακροδέκτες Εξόδου

modus_operandi ▷ Έξοδος που καθορίζει τη λειτουργία του συστήματος ως πομπό ('0') ή ως δέκτη ('1')

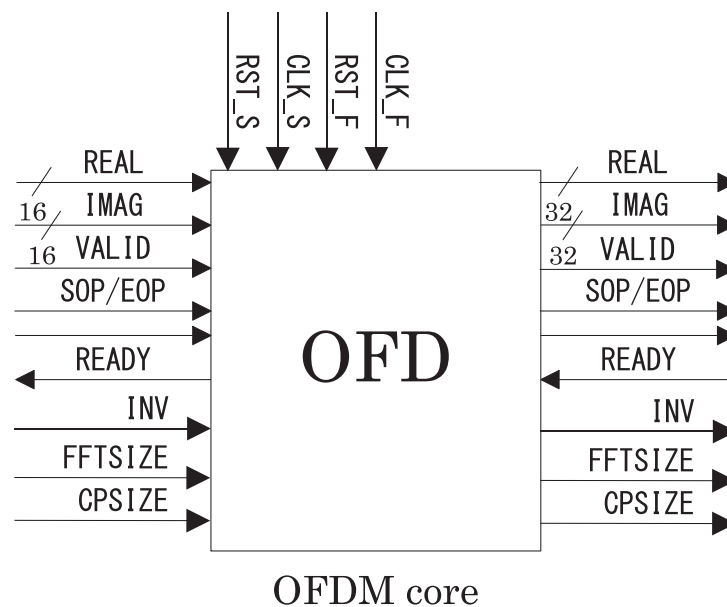
modus_init ▷ Ενεργοποιείται άπαξ μία φορά κατά την έναρξη κάθε TDD slot, ώστε να υποστηρίξει λειτουργίες αρχικοποίησης.

Η λειτουργία του TDD controller είναι να συντονίζει τη λειτουργία πομπού και δέκτη. Το κύκλωμα αποτελείται από δύο datapath, το TX και το RX, τα οποία μπορούν να λειτουργούν παράλληλα. Ωστόσο ένα υποσύστημα (Altera OFDM Core) είναι κοινόχρηστο και στα δύο datapath. Βασική λειτουργία του TDD controller είναι η ολοκλήρωση του εκάστοτε datapath, ελέγχοντας τον πολυπλέκτη εισόδου και τον αποπολυπλέκτη εξόδου του OFDM core.

Η λειτουργία simplex επιτυγχάνεται με την καθήλωση στο λογικό '0' της εξόδου *modus_operandi*. Σε αυτή, το κύκλωμα λειτουργεί συνεχώς ως πομπός ή ως δέκτης και ο TDD controller, ενεργοποιεί είτε το TX, είτε το RX datapath. Για την αλλαγή λειτουργίας, απαιτείται να δημιουργηθεί νέο "στιγμιότυπο" του κυκλώματος.

Στο παρόν κύκλωμα δεν έχει υλοποιηθεί η πλήρης λειτουργία half duplex καθώς είναι εκτός των προδιαγραφών. Ωστόσο, σε μία μελλοντική επέκταση, ο TDD controller θα υλοποιούσε τη λειτουργία της πολυπλεξίας διαίρεσης χρόνου. Εδώ, το κάθε datapath λειτουργεί για ένα μικρό μη επικαλυπτόμενο χρονικό διάστημα (TDD slot). Ο TDD controller είναι υπευθυνος για τον χρονισμό της διαδικασίας της πολυπλεξίας, ενεργοποιώντας το κάθε datapath τη σωστή στιγμή.

3.2.3 Το module Altera OFDM Core



Σχήμα 3.10: Διάγραμμα Ακροδεκτών Altera OFDM Core

Στο Σχήμα 3.10 παρουσιάζεται το διάγραμμα ακροδεκτών του Altera OFDM Core. Πρόκειται για τον "χωροταξικό" και λειτουργικό πυρήνα του συστήματος. Οι βασικές αρχές που διέπουν τη λειτουργία του είναι:

- η προσθήκη της κυκλικής επέκτασης
- ο μετασχηματισμός FFT και IFFT
- η αφαίρεση της κυκλικής επέκτασης

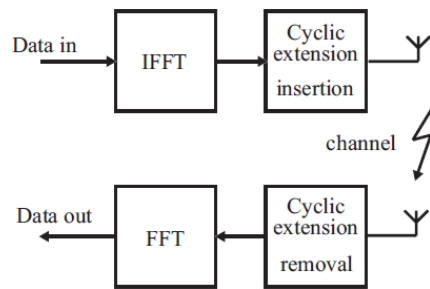
και απεικονίζονται στο σχήμα 3.11.

Είναι κοινόχρηστο module, δηλαδή είναι παρον τόσο στο datapath του πομπού όσο και του δέκτη. Η λειτουργία του διαφοροποιείται ανάλογα με πιο datapath είναι ενεργό κάθε φορά (είτε πρόκειται για Half-Duplex τρόπο μετάδοσης, είτε για simplex). Στο διάγραμμα 3.12 απεικονίζονται τα βασικά συστατικά του και η θέση του στο συνεπτυγμένο RXTX datapath.¹⁷

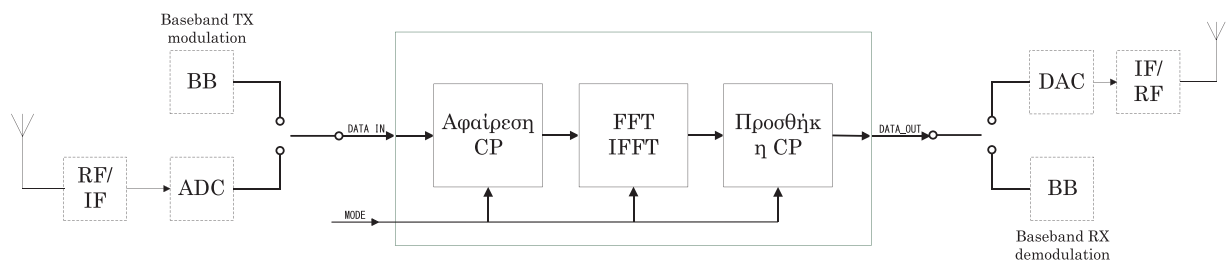
Τώρα, πυρήνας της διαμόρφωσης OFDM είναι ο IFFT και αντίστοιχα της αποδιαμόρφωσης ο FFT. Πρόκειται για πολύ απαιτητικές, υπολογιστικά, λειτουργίες που υλοποιούνται από το Altera FFT MegaFunction.¹⁸

¹⁷ Δεν απεικονίζεται το υπόλοιπο σύστημα, ενώ οι όροι DDC και DUC είναι ακρωνύμια των *Digital Down Conversion* και *Digital Up Conversion*, αντίστοιχα. Στο συγκεκριμένο παράδειγμα είναι ενεργοποιημένο το RX datapath.

¹⁸ Τα Altera MegaFunction είναι υλοποιημένα κυκλώματα απο την Altera, βελτιστοποιημένα για τα FPGA της συγκεκριμένης



Σχήμα 3.11: Βασική αρχή λειτουργίας του OFDM Core



Σχήμα 3.12: Κοινοχρησία του module OFDM Core

Προκειμένου να επιτευχθεί εξοικονόμηση υλικού, το FFT module δέχεται και δεύτερο ρολόι (διπλάσιας συχνότητας από το βασικό ρολόι, που είναι διαθέσιμο στα υπόλοιπα module) και επαναχρησιμοποιείται.

Η ορθότητα της κοινής χρήσης εξασφαλίζεται με τη σωστή πολυπλεξία των σχετικών σημάτων. Το σήμα ελέγχου *inv_in* που προέρχεται από τον *tdd controller* καθορίζει τη χρήση του FFT module ως εξής :

- *inv_in* = '1' ⇒ IFFT @ TX Path
- *inv_in* = '0' ⇒ FFT @ RX Path

Επιπλέον, τα module *CP-Removal* και *CP-Insertion* τα οποία αφαιρούν και εισάγουν, αντίστοιχα, το κυκλικό πρόθεμα στο σήμα έχουν γνώση του τρόπου λειτουργίας, δηλαδή αν πρόκειται για εκπεμπόμενα ή ληφθέντα δεδομένα και δρουν αναλόγως. Περισσότερες πληροφορίες για την αρχιτεκτονική και του module OFDM core, μπορούν να βρεθούν στο [Ofd].

```
entity ofdm_int is
  generic
  (
    DFFTOUTWIDTH : natural;
    DFFTINWIDTH  : natural;
    DOUTWIDTH    : natural;
    MWIDTH       : natural;
    MADDR_WIDTH_D : natural; -- double buffer depth
    MADDR_WIDTH_S : natural; -- single buffer depth
    CPWIDTH      : natural;
    NWIDTH       : natural
  );
  port
  (
```

εταιρίας. Παρέχονται με τη μορφή παραμετροποιήσιμου IP core. Δηλαδή δεν έχει κανείς πρόσβαση στον πηγαίο κώδικα της υλοποίησής τους, αλλά έχει αρκετούς βαθμούς ελευθερίας για τον προσδιορισμό της συμπεριφοράς τους. Μαζί με το περιβάλλον σύνθεσης Quartus παρέχεται μία πληθώρα megafunction για διάφορες λειτουργίες.

```

clk_f      : in std_logic;
clk_s      : in std_logic;
rst_f_n    : in std_logic;
rst_s_n    : in std_logic;
-- Control
cpsize_in  : in std_logic_vector (CPWIDTH - 1 downto 0);
cpsize_out : out std_logic_vector (CPWIDTH - 1 downto 0);
fftsize_in : in std_logic_vector (NWIDTH - 1 downto 0);
fftsize_out : out std_logic_vector(NWIDTH -1 downto 0);
inv_in     : in std_logic;          -- '1' for IFFT
inv_out    : out std_logic;
-- Avalon Streaming Data Sink Input Interface
din_valid  : in std_logic;
din_real   : in std_logic_vector (DFFFTINWIDTH - 1 downto 0);
din_imag   : in std_logic_vector (DFFFTINWIDTH - 1 downto 0);
din_sop    : in std_logic;
din_eop    : in std_logic;
din_ready  : out std_logic;
-- Avalon Streaming Source Interface
dout_ready : in std_logic;
dout_valid : out std_logic;
dout_sop   : out std_logic;
dout_eop   : out std_logic;
dout_real  : out std_logic_vector (DOUTWIDTH - 1 downto 0);
dout_imag  : out std_logic_vector (DOUTWIDTH - 1 downto 0)
);
end ofdm_int;

```

Ακολουθεί μια συνοπτική περιγραφή του κάθε ακροδέκτη :

Ακροδέκτες Εισόδου

clk_f ▷ Το γρήγορο ρολόι για το ODFM_INT MegaCore.

clk_s ▷ Το βασικό ρολόι του συστήματος.

rst_f_n ▷ Το σήμα επαναφοράς που αντιστοιχεί στο γρήγορο ρολόι, αρνητικής λογικής.

rst_s_n ▷ Το βασικό σήμα επαναφοράς του συστήματος, αρνητικής λογικής.

cpsize_in ▷ Είσοδος που καθορίζει το μήκος του διαστήματος φύλαξης για το τρέχον σύμβολο.

fftsize_in ▷ Είσοδος που καθορίζει το μέγεθος του ζεύγους FFT/IFFT για το τρέχον σύμβολο.

inv_in ▷ Είσοδος που επιλέγει το είδος του μετασχηματισμού : FFT(0) ή IFFT(1).

din_valid ▷ Είσοδος που διεκπεραιώνει το ρόλο *sink valid* του Avalon ST Interface.

din_real ▷ Πραγματικό μέρος της ακολουθίας εισόδου (16 bit signed). Μπορεί να είναι samples στο χρόνο ή στη συχνότητα. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

din_imag ▷ Φανταστικό μέρος της ακολουθίας εισόδου (16 bit signed). Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

din_sop ▷ Είσοδος που διεκπεραιώνει το ρόλο *startofpacket* του Avalon ST Interface.

din_eop ▷ Είσοδος που διεκπεραιώνει το ρόλο *endofpacket* του Avalon ST Interface.

dout_ready ▷ Είσοδος που διεκπεραιώνει το ρόλο *source ready* του Avalon ST Interface.

Ακροδέκτες Εξόδου

cpsize_out ▷ Έξοδος που περιέχει το μήκος του διαστήματος φύλαξης για το τρέχον σύμβολο.

fftsize_out ▷ Έξοδος που περιέχει το μέγεθος του ζεύγους FFT/IFFT για το τρέχον σύμβολο.

inv_out ▷ Έξοδος που περιέχει το είδος του μετασχηματισμού : FFT(0) ή IFFT(1) για το τρέχον σύμβολο.

dout_valid ▷ Έξοδος που διεκπεραιώνει το ρόλο *source valid* του Avalon ST Interface.

dout_sop ▷ Έξοδος που διεκπεραιώνει το ρόλο *startofpacket* του Avalon ST Interface.

dout_eop ▷ Έξοδος που διεκπεραιώνει το ρόλο *endofpacket* του Avalon ST Interface.

dout_real ▷ Πραγματικό μέρος της ακολουθίας εξόδου (32 bit signed). Μπορεί να είναι samples στη συχνότητα ή στο χρόνο. Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

dout_imag ▷ Φανταστικό μέρος της ακολουθίας εξόδου (32 bit signed). Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

din_ready ▷ Έξοδος που διεκπεραιώνει το ρόλο *sink ready* του Avalon ST Interface.

3.2.4 Σχολιασμός της μετα-FFT Κλιμάκωσης (*Post-FFT Scaling*)

Στα samples που εξέρχονται του OFDM Core module, τόσο στο TX, όσο και στο RX path, υπάρχουν δύο ολισθητές που εφαρμόζουν κλιμάκωση στα δεδομένα εξόδου του. Η ύπαρξή τους έχει να κάνει με την εσωτερική υλοποίηση το ζεύγους του μετασχηματισμού FFT/IFFT στο FFT MegaCore της Altera. Για να εξηγηθεί καλύτερα η κλιμάκωση, επαναλαμβάνεται εδώ η μαθηματική περιγραφή της διαμόρφωσης OFDM στο διακριτό χρόνο. Το διαμορφωμένο σήμα είναι :¹⁹

$$z(n) = \sum_{k=0}^{N-1} S_k \exp(j2\pi k \frac{n}{N}) = IDFT\{S_k\}$$

ενώ το αποδιαμορφωμένο :

$$G_k = \frac{1}{N} \sum_{n=0}^{N-1} z(n) \exp(-j2\pi k \frac{n}{N}) = DFT\{z(n)\} = Z(k) = DFT\{IDFT\{S_k\}\} = S_k$$

Για τις rms²⁰ τιμές των πλατών των παραπάνω σημάτων ισχύει:

- Στον πομπό, η rms τιμή του πλάτους της εξόδου του IFFT $z(\tilde{n})$ είναι \sqrt{N} φορές μεγαλύτερη από αυτή της εισόδου \tilde{S}_k .
- Στο δέκτη η rms τιμή του πλάτους της εξόδου του FFT, \tilde{G}_k είναι $\frac{1}{\sqrt{N}}$ φορές μικρότερη από αυτή της εισόδου $z(\tilde{n})$, λόγω του συντελεστή $1/N$.
- Τελικά ισχύει $\tilde{G}_k = \tilde{S}_k$. Αυτή είναι η σχέση που εξασφαλίζει τη μαθηματική ορθότητα του συστήματος.
- Γενικά, τόσο στον FFT όσο και στον IFFT αν αγνοηθεί ο συντελεστής $1/N$, το σήμα εξόδου τους έχει rms τιμή \sqrt{N} φορές μεγαλύτερη από το σήμα εισόδου. Αυτό είναι μια εγγενής ιδιότητα του μετασχηματισμού.

Η ύπαρξη του παράγοντα $\frac{1}{N}$ δεν προβλέπεται στο Altera OFDM MegaCore και ως εκ τούτου θα πρέπει να υλοποιηθεί εξωτερικά. Χωρίς το συντελεστή, η \tilde{G}_k θα ήταν N φορές μεγαλύτερη από την \tilde{S}_k . Ένας πιθανός τρόπος επίλυσης του προβλήματος θα ήταν η εισαγωγή του συντελεστή $1/N$ μετά τον FFT στο δέκτη, όπως συμβολίζεται στους παραπάνω τύπους.

Ωστόσο υπάρχει ένα άλλο ζήτημα, τεχνικής φύσεως, που επιβάλλει μία διαφορετική προσέγγιση στο θέμα της κλιμάκωσης. Λόγω της αυξημένης rms τιμής της εξόδου του μετασχηματισμού FFT/IFFT, ο τύπος της εξόδου του FFT MegaCore είναι int32 (ενώ της εισόδου του int16). Αυτό σημαίνει πρακτικά, πως τα post

¹⁹(S2.1.5) Για την παρούσα ανάλυση δεν είναι αναγκαία η υπόθεση ότι πέρασε από κάποιο κανάλι.

²⁰Η rms τιμή του πλάτους ενός σήματος $S(n)$ υπολογίζεται ως η ρίζα της μέσης τετραγωνικής τιμής του (*root mean square*). Ισχύει δηλαδή $\tilde{S} = \sqrt{E_n[|S(n)|^2]}$. Επιπλέον, η μέση ισχύς του σήματος είναι $P_S = E[|S(n)|^2] = \tilde{S}^2$. Συνεπώς, το σταθερό σήμα $S'(n) = \tilde{S}$, ίσης διάρκειας με το $S(n)$, έχει την ίδια μέση ισχύ με αυτό.

FFT modules στο TX και RX path πρέπει να υλοποιηθούν σε αριθμητική int32. Ωστόσο η διακριτική ανάλυση των DAC/ADC που χρησιμοποιούνται συνήθως, είναι στα 16bit ή και λιγότερο. Γι' αυτό το λόγο, έχει επιλεγεί ως προδιαγραφή εύρους των DAC/ADC τα 16 bit. Επιπλέον ο τύπος int32 για τα post FFT modules στο δέκτη, εισάγει επιπλέον υπολογιστικό φόρτο σε σχέση με τον int16, χωρίς να προσφέρει ιδιαίτερα σε ακρίβεια. Συνεπώς τα post FFT samples μετατρέπονται σε τύπο int16.

Για να γίνει σωστά η μετατροπή, αρχικά πρέπει να γίνει κλιμάκωση του πλάτους της εξόδου, ώστε η rms τιμή του να έρθει στα ίδια επίπεδα με αυτή της εισόδου. Η κλιμάκωση των post FFT samples μπορεί να γίνει με οποιοδήποτε συντελεστή αρκεί το ολικό γινόμενο των συντελεστών πομπού και δέκτη να ισούται με $1/N$. Η βέλτιστη επιλογή συντελεστών είναι $1/\sqrt{N}$ στον πομπό και $1/\sqrt{N}$ στον δέκτη.

Για πρακτικούς λόγους επιλέχτηκε η υλοποίηση του γινομένου $\frac{1}{N1} \times \frac{1}{N2} = \frac{1}{N}$, με τον πρώτο όρο $\frac{1}{N1}$ να υλοποιείται στην πλευρά του TX και το δεύτερο $\frac{1}{N2}$ να υλοποιείται στην πλευρά του RX.

Σε κάθε περίπτωση, οι όροι $N1, N2$ (και φυσικά ο N) είναι δυνάμεις του 2, δηλαδή $2^{F_{TX}} = N1$ και $2^{F_{RX}} = N2$, όπου $F_{TX}, F_{RX} \in \mathbb{N}$. Για αυτό το λόγο, οι διαιρέτες έχουν υλοποιηθεί κυκλωματικά ως απλοί ολισθητές (shifters). Στον TX υπάρχει ολίσθηση κατά F_{TX} προς τα δεξιά ενώ στον RX υπάρχει ολίσθηση κατά F_{RX} προς τα δεξιά. Με αυτό τον τρόπο αποκαθίσταται η μαθηματική εγκυρότητα κατά την διαμόρφωση και αποδιαμόρφωση (κατά OFDM) των δεδομένων επιτυγχάνοντας, παράλληλα, εξοικονόμηση υλικού.

Μετά την κλιμάκωση ακολουθεί η καθαυτό μετατροπή σε int16 με saturation. Το αποτέλεσμα είναι ο ψαλιδισμός του σήματος στα σημεία που εξακολουθεί να βρίσκεται σε υψηλά επίπεδα. Το φαινόμενο αυτό πρέπει να αποφεύγεται καθώς εισάγει αρμονική παραμόρφωση στο σήμα εξόδου.

3.2.5 Σχολιασμός της προ-FFT Κλιμάκωσης (Pre FFT scaling)

Για την αποφυγή του clipping στα post FFT samples τόσο στον πομπό, όσο και στο δέκτη, εισάγεται η έννοια της pre FFT κλιμάκωσης. Το clipping συμβαίνει στα samples με υψηλή τιμή που βρίσκεται εκτός του δυναμικού εύρους του τύπου αποθήκευσής τους (int16). Παρόλο που οι συντελεστές post FFT κλιμάκωσης έχουν επιλεγεί ώστε να ισοκατανέμουν, όσο είναι δυνατό, την rms τιμή του πλάτους πριν και μετά τον FFT, η υψηλές τιμές του PAPR²¹ στο σήμα εξόδου, επιβάλλουν ένα σχετικά χαμηλό άνω όριο στην rms τιμή της. Το όριο αυτό μεταφέρεται και στην είσοδο. Ο ρόλος της pre FFT κλιμάκωσης είναι ακριβώς το να μη ξεπεραστεί αυτό το όριο.

Στον πομπό η pre IFFT κλιμάκωση γίνεται από το συντελεστή κλιμάκωσης των αστερισμών²² και ενσωματώνεται στην αποθηκευμένη μορφή τους. Ρυθμίζεται κατά τη σχεδίαση μέσω του μηχανισμού asset-parameter dispatching.

Στον δέκτη το ρόλο της pre FFT κλιμάκωσης παίζει ένας συντελεστής κέρδους (gain factor), είτε στο αναλογικό τμήμα πριν τον ADC, είτε στο ψηφιακό μετά τον ADC. Η τιμή του πρέπει να προρυθμιστεί έτσι ώστε η rms τιμή των samples εισόδου στον FFT να βρίσκεται στο προκαθορισμένο επίπεδο. Στο παρόν σύστημα δεν υπάρχει τέτοιος συντελεστής, καθώς θεωρείται πως το σήμα εισόδου στο δέκτη, βρίσκεται στη σωστή στάθμη.²³ Σε πολλά πραγματικά συστήματα υπάρχει ο μηχανισμός του Automatic Gain Control (AGC), ώστε να ρυθμίζει αυτόματα την rms τιμή του σήματος εισόδου στην επιθυμητή τιμή. Αυτό είναι απαραίτητο καθώς η στάθμη του σήματος εισόδου εξαρτάται από τις συνθήκες του καναλιού και δεν μπορεί να προκαθοριστεί επακριβώς.

Στην παρούσα υλοποίηση οι τιμές των συντελεστών pre FFT κλιμάκωσης έχουν επιλεγεί μετά από δοκιμαστικές εκπομπές και λήψεις για διάφορους τύπους καναλιών. Κάθε φορά ελαμβάνετο η μέση τιμή του PAPR και ο αριθμός των clipped samples. Οι συντελεστές επιλέχτηκαν με συντηρητικό τρόπο, έτσι ώστε να αποφευχθεί πλήρως το clipping.

²¹Για μία συζήτηση πάνω στο PAPR βλέπε §2.1.2

²²CONSTEL_SCALE_INT16MAX_PRCNT

²³ Η pre FFT κλιμάκωση στο δέκτη, γίνεται κατά την προσομοίωση του συστήματος στα υποστηρικτικά script MATLAB. Περισσότερα για αυτό αναφέρονται στο κεφάλαιο 5.

Σημειώνεται τέλος, πως η παραπάνω προσέγγιση αφορά την προστασία από clipping στους DAC και ADC μόνο. Τα προβλήματα που προκαλεί η παραμόρφωση -λόγω υψηλού PAPR- του σήματος στις αναλογικές βαθμίδες εξακολουθούν να υφίστανται και λύνονται με άλλους τρόπους, που είναι εκτός των σκοπών της παρούσας υλοποίησης.

3.3 Δομικές Μονάδες Μέτρησης της Πληροφορίας

Πριν την περιγραφή του συστήματος, είναι απαραίτητη η παρουσίαση της ορολογίας που χρησιμοποιείται για την αναφορά στις δομές λογικής οργάνωσης του ποσού και του είδους της πληροφορίας που επεξεργάζεται η κάθε μονάδα του συστήματος.

Υπάρχουν δύο κατηγορίες λογικής οργάνωσης δεδομένων: Από τη σκοπιά του interface διασύνδεσης των υποσυστημάτων (συστημική οργάνωση) και από τη σκοπιά των λειτουργιών του τηλεπικοινωνιακού συστήματος (λειτουργική οργάνωση).

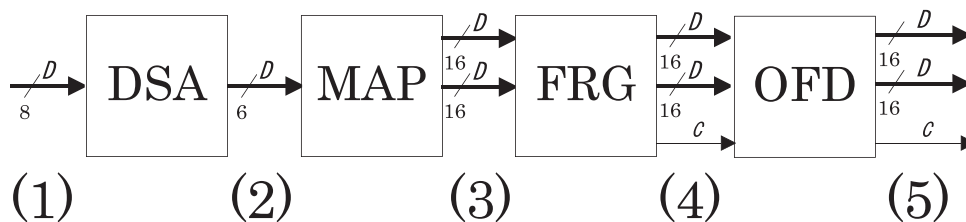
isymbol → Ελάχιστο κβάντο πληροφορίας που μεταφέρεται σε κάθε data cycle ανά θύρα δεδομένων, στην ενδοσυστημική επικοινωνία μεταξύ των υπομονάδων (modules). Αποτελείται από μία λέξη σταθερού -ανά υπομονάδα- μήκους.

unit → Το σύνολο των isymbol που μεταφέρονται σε κάθε data cycle μεταξύ δύο υποσυστημάτων. Ένα υποσύστημα μπορεί να έχει πολλές θύρες δεδομένων.

ipacket → Υψηλότερη ιεραρχικά μονάδα πληροφορίας που αποτελείται από μία ομάδα διαδοχικών isymbols, όπως αυτά οριοθετούνται από τα σήματα *sop* & *eop*, τα οποία ενσαρκώνουν τους ρόλους *startofpacket* και *endofpacket*. Η μεταφορά ενός ipacket διαρκεί πολλά data cycles.

Αλυσίδα Πομπού

Στο διάγραμμα 3.13 παρουσιάζεται η αλυσίδα του πομπού, όπου έχει δοθεί έμφαση στα εύρη των θυρών που επιτελούν το ρόλο *sink/source data* του Avalon ST Interface. Πρόκειται δηλαδή για τις θύρες μετάδοσης και λήψης του καθαυτού σήματος πληροφορίας και όχι των control σημάτων που συντονίζουν τη λειτουργία του συστήματος.



Σχήμα 3.13: Συνεπτυγμένη αλυσίδα πομπού

Ακολουθεί η αντιστοίχιση συστημικών \Leftrightarrow (λειτουργικών) τηλεπικοινωνιακών μεγεθών, όπου αυτό έχει κάποιο νόημα.

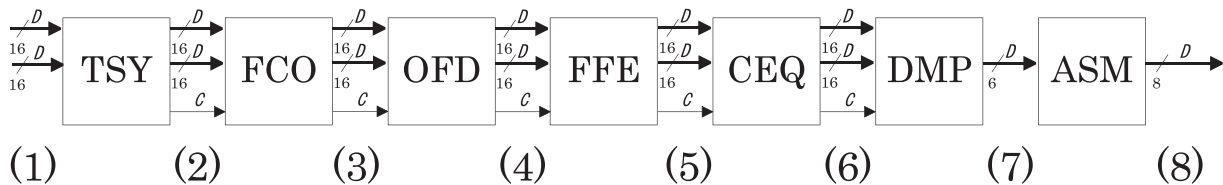
1. isymbol \Leftrightarrow byte
2. isymbol \Leftrightarrow k-blocks²⁴

²⁴όπου $k = \text{subsymbol_block_size} = \log_2 Q, Q_{QAM} = \{4, 16, 64\}$

3. iunit (1 ανά subcarrier) \Leftrightarrow subsymbol (σημείο αστερισμού QAM)
4. iunit \Leftrightarrow subsymbol • ipacket \Leftrightarrow ωφέλιμο OFDM σύμβολο (στη συχνότητα)
5. iunit \Leftrightarrow complex time sample • ipacket \Leftrightarrow OFDM symbol (με κυκλικό πρόθεμα, στο χρόνο)

Αλυσίδα Δέκτη

Σε πλήρη αντιστοιχία με τον πομπό, στο διάγραμμα 3.14 παρουσιάζονται τα αντίστοιχα εύρη των θυρών *data* για τα modules του δέκτη.



Σχήμα 3.14: Συνεπτυγμένη αλυσίδα δέκτη

Και εδώ, ακολουθεί η αντιστοίχιση συστημικών μεγεθών σε τηλεπικοινωνιακά, όπου αυτό κρίνεται απαραίτητο.

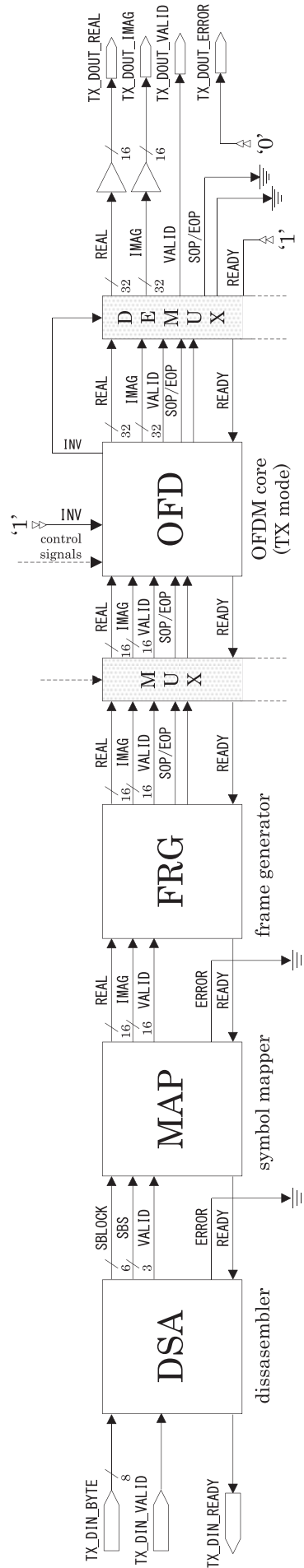
1. iunit \Leftrightarrow complex time sample
2. iunit \Leftrightarrow complex time sample • ipacket \Leftrightarrow OFDM symbol
3. iunit \Leftrightarrow complex time sample • ipacket \Leftrightarrow OFDM symbol
4. iunit \Leftrightarrow subsymbol (ανά subcarrier) • ipacket \Leftrightarrow ωφέλιμο OFDM symbol, στη συχνότητα
5. iunit \Leftrightarrow subsymbol (ανά subcarrier) • ipacket \Leftrightarrow ωφέλιμο OFDM symbol, στη συχνότητα
6. iunit \Leftrightarrow subsymbol (ανά subcarrier) • ipacket \Leftrightarrow ωφέλιμο OFDM symbol, στη συχνότητα
7. isymbol \Leftrightarrow k-block
8. isymbol \Leftrightarrow byte

3.4 Αλυσίδα Υποσυστήματος Πομπού

3.4.1 Η Διαδρομή Δεδομένων στον Πομπό (TX Datapath)

Στο σχήμα 3.15 παρουσιάζεται η πλήρης αλυσίδα του πομπού. Τα κύρια modules που τη συναποτελούν είναι τα :

- *Disassembler* ή Τεμαχιστής
- *Symbol Mapper* ή Διαμορφωτής QAM
- *Frame Generator* ή Παραγωγός Πλαισίων
- *OFDM Modulator* ή Διαμορφωτής OFDM



Σχήμα 3.15: Το TX Datapath

Συνοπτική Περιγραφή της Πορείας της Πληροφορίας στον Πομπό

Παραδοχές Τα δεδομένα πληροφορίας βρίσκονται σε ψηφιακή μορφή. Μπορεί να αποτελούν κάποιο αρχείο στον υπολογιστή ή μέσω ADC να έχουν περάσει στο σύστημα.

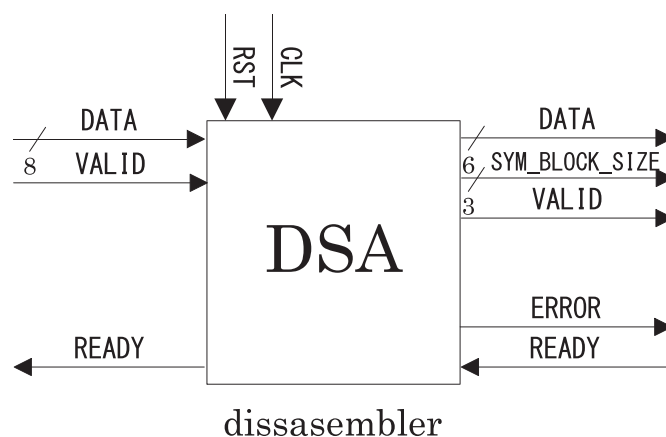
Σε πρώτο στάδιο, διαβάζονται ανά byte και διαβιβάζονται στον *disassembler* όπου και τεμαχίζονται σε συρμούς των 2, 4 ή 6 bit, ανάλογα με το είδος της επιθυμητής διαμόρφωσης QAM. Κατόπιν συνεχίζουν στον *symbol mapper* όπου και λαμβάνει χώρα η αντιστοίχιση των συρμών σε σύμβολα του "αλφαβήτου" QAM. Επόμενο σταθμό αποτελεί ο *frame generator*, όπου παρεμβάλλονται στη ροή των δεδομένων τα - απαραίτητα για το συγχρονισμό και εκτίμηση καναλιού στο δέκτη - training symbols. Κατόπιν τα δεδομένα διαβιβάζονται στο *OFDM_Modulator* όπου και γίνεται η εισαγωγή του cyclic prefix και λαμβάνει χώρα το πέρασμα από τον IFFT, ολοκληρώνοντας, έτσι, τη διαμόρφωση OFDM. Τελευταίο²⁵ στάδιο είναι η κλιμάκωση (scaling) των δεδομένων.

3.4.2 Συστημική Περιγραφή Μονάδων

Σε αυτό το τμήμα παρουσιάζονται, με κάπως περισσότερη λεπτομέρεια, οι υπομονάδες (modules) του συστήματος. Ο σκελετός της παρουσίασης είναι κοινός για όλα τα modules και αποτελείται από ένα διάγραμμα που απεικονίζει τους ακροδέκτες I/O, ένα τμήμα κώδικα VHDL που αποτελεί τη δήλωση οντότητας (entity declaration), μια συνοπτική περιγραφή των ακροδεκτών και των σημάτων που σχετίζονται με αυτούς. Τέλος, την παρουσίαση κάθε module κλείνει μία λεκτική περιγραφή των λειτουργιών που αυτό επιτελεί, χωρίς όμως αυτή να εξαντλεί το πως είναι δομημένο αρχιτεκτονικά το module, καθώς το Κεφάλαιο 4 είναι αφιερωμένο σε αυτό το σκοπό. Με άλλα λόγια, σε αυτό το κεφάλαιο δίνεται έμφαση στο *τί* και στο *γιατί* και όχι στο *πώς*.

Disassembler

Στο Σχήμα 3.16 παρουσιάζεται το διάγραμμα ακροδεκτών του Disassembler και ακολουθεί μια συνοπτική περιγραφή του κάθε ακροδέκτη.



Σχήμα 3.16: Διάγραμμα ακροδεκτών Disassembler

```
entity dis_asm is
  generic (
```

²⁵Σε πραγματικές συνθήκες ακολουθεί η μίξη των συνιστωσών I,Q στα IF, το ανέβασμα στις συχνότητες RF και η εκπομπή από την κεραία.

```

    D_IN_WIDTH : natural;
    D_OUT_WIDTH : natural
);
port (
    clk          : in std_logic;
    rst          : in std_logic;
    -- Avalon Streaming Input (Sink) Interface
    in_data      : in std_logic_vector (D_IN_WIDTH-1 downto 0);
    in_valid     : in std_logic;
    in_ready     : out std_logic;
    -- Avalon Streaming Output (Source) Interface
    out_ready    : in std_logic;
    out_valid    : out std_logic;
    out_data     : out std_logic_vector (D_OUT_WIDTH-1 downto 0);
    out_error    : out std_logic;
    -- SBS Output Spread
    sym_block_size : out std_logic_vector(2 downto 0)
);
end dis_asm;

```

Ακροδέκτες Εισόδου

clk ▷ Το βασικό ρολόι του συστήματος.

rst ▷ Το βασικό σήμα επαναφοράς του συστήματος.

in_data ▷ Είσοδος δεδομένων προς μετάδοση, εύρους 8 bit (1 byte). Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_valid ▷ Είσοδος που διεκπεραιώνει το ρόλο *sink valid* του Avalon ST Interface.

out_ready ▷ Είσοδος που διεκπεραιώνει το ρόλο *source ready* του Avalon ST Interface.

Ακροδέκτες Εξόδου

out_data ▷ Έξοδος δεδομένων σταθερού εύρους 6 bit, με ωφέλιμο εύρος 2, 4 ή 6 bit (ανά QAM 4,16 ή 64 αντίστοιχα). Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_valid ▷ Έξοδος που διεκπεραιώνει το ρόλο *source valid* του Avalon ST Interface.

out_error ▷ Έξοδος που διεκπεραιώνει το ρόλο *error* του Avalon ST Interface.

sym_block_size ▷ Έξοδος εύρους 3 bit, περιέχει την πληροφορία του ωφέλιμου εύρους της θύρας *out_data*. Ουσιαστικά, κωδικοποιεί τον τύπο διαμόρφωσης QAM που θα λάβει χώρα στο επόμενο module. Αναφέρεται στα τρέχοντα περιεχόμενα της εξόδου *out_data*.

in_ready ▷ Έξοδος που διεκπεραιώνει το ρόλο *sink ready* του Avalon ST Interface.

Η κύρια λειτουργία του disassembler συνίσταται στην κατάτμηση του byte εισόδου σε block μεταβλητού εύρους των k bit (k -blocks). Είναι ένα module που μετά από αρχική καθυστέρηση (delay) 2 κύκλων, σε κάθε επόμενο χτύπο ρολογιού εμφανίζει στην έξοδο δεδομένων του και από ένα k -block. Το εκάστοτε k -block "συνοδεύεται" από την πληροφορία Q , της διαμόρφωσης QAM που του αναλογεί, που είναι διαθέσιμη στην έξοδο *sym_block_size*. Η πληροφορία Q κωδικοποιείται ως το εύρος k , του k -block, σε αναπαράσταση *unsigned integer*. Η σχέση μεταξύ k και Q είναι $k = \log_2(Q)$. Εφόσον οι διαθέσιμοι τύποι διαμόρφωσης κατά QAM είναι $Q \in \{4, 16, 64\}$, τότε οι πιθανές τιμές για το εύρος του block είναι $k \in \{2, 4, 6\}$.

Όπως αναφέρθηκε, το ωφέλιμο εύρος των k -block είναι μεταβλητό ανά *isymbol*. Ωστόσο, το κάθε k -block κωδικοποιείται ως μία λέξη σταθερού εύρους 6 bit, σύμφωνα με την πιο απαιτητική περίπτωση. Η κωδικοποίηση του k -block έχει ως εξής :

$$block = [X]\&Payload$$

όπου X είναι τα bit πλεονασμού (αδιάφορα bit). Το πλήθος τους N_X , μπορεί να πάρει τις τιμές $\{0, 2, 4\}$ σύμφωνα με τη σχέση :

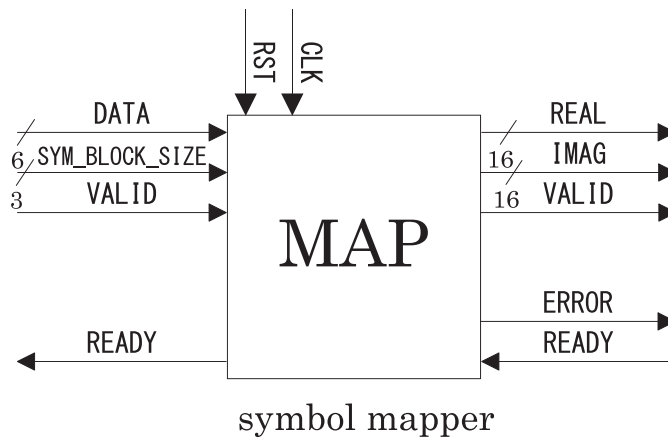
$$N_X = \log_2 \frac{64}{Q}, Q = \{64, 16, 4\}$$

Η εκάστοτε τιμή Q προέρχεται από το μοτίβο διαμόρφωσης QAM (mod_pattern), που είναι γνωστό στον disassembler.

Ο disassembler είναι ένα module που ασκεί backpressure στο upstream, καθώς κάθε isymbol εισόδου, μπορεί να παράγει ένα ή περισσότερα isymbol εξόδου. Για τον disassembler που είναι το πρώτο module στην αλυσίδα του πομπού, upstream θεωρείται η διεργασία που παρέχει τα byte στην είσοδο δεδομένων του. Δεδομένου, τώρα, ότι μετά από αρχικό delay 2 κύκλων υπάρχει συνεχής ροή block, ο disassembler δεν ασκεί frontpressure στο downstream και δεν εισάγει φουσαλλίδες καθυστέρησης στο datapath.

Symbol Mapper

Στο Σχήμα 3.17 παρουσιάζεται το διάγραμμα ακροδεκτών του Symbol Mapper και ακολουθεί μια συνοπτική περιγραφή του κάθε ακροδέκτη.



Σχήμα 3.17: Διάγραμμα ακροδεκτών Symbol Mapper

```
entity sym_map is
  generic (
    KINPUT_MAX : natural;
    DATA_OUT_WIDTH : natural
  );
  port (
    clk : in std_logic;
    rst : in std_logic;
    sym_block_size : in std_logic_vector(2 downto 0);
    -- Avalon Streaming Input (Sink) Interface
    in_data : in std_logic_vector (KINPUT_MAX-1 downto 0);
    in_valid : in std_logic;
    in_ready : out std_logic;
    -- Avalon Streaming Output (Source) Interface
    out_ready : in std_logic;
    out_valid : out std_logic;
    out_real : out std_logic_vector (DATA_OUT_WIDTH-1 downto 0);
    out_imag : out std_logic_vector (DATA_OUT_WIDTH-1 downto 0);
    out_error : out std_logic
  );
end sym_map;
```

Ακροδέκτες Εισόδου

clk ▷ Το βασικό ρολόι του συστήματος.

rst ▷ Το βασικό σήμα επαναφοράς του συστήματος.

in_data ▷ Είσοδος δεδομένων εύρους 6 bit, με ωφέλιμο εύρος 2, 4 ή 6 bit (ανά QAM 4,16 ή 64 αντίστοιχα). Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_valid ▷ Είσοδος που διεκπεραιώνει το ρόλο *sink valid* του Avalon ST Interface.

sym_block_size ▷ Είσοδος εύρους 3 bit, περιέχει την πληροφορία του τύπου διαμόρφωσης QAM που θα λάβει χώρα. Αναφέρεται στα τρέχοντα περιεχόμενα της εισόδου.

out_ready ▷ Είσοδος που διεκπεραιώνει το ρόλο *source ready* του Avalon ST Interface.

Ακροδέκτες Εξόδου

out_valid ▷ Έξοδος που διεκπεραιώνει το ρόλο *source valid* του Avalon ST Interface.

out_real ▷ Έξοδος δεδομένων εύρους 16 bit. Περιέχει την πραγματική συνιστώσα του μιγαδικού σήματος που προκύπτει μετά τη διαμόρφωση κατά QAM X, όπου $X = [\text{sym_block_size}]$. Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_imag ▷ Έξοδος δεδομένων εύρους 16 bit. Περιέχει τη φανταστική συνιστώσα του μιγαδικού σήματος που προκύπτει μετά τη διαμόρφωση κατά QAM X, όπου $X = [\text{sym_block_size}]$. Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_error ▷ Έξοδος που διεκπεραιώνει το ρόλο *error* του Avalon ST Interface.

in_ready ▷ Έξοδος που διεκπεραιώνει το ρόλο *sink ready* του Avalon ST Interface.

Η βασική λειτουργία του symbol mapper είναι η διαμόρφωση κατά QAM του σήματος πληροφορίας. Οι δυνατές διαμορφώσεις είναι οι : QAM4 (QPSK), QAM16 και QAM64. Από πλευράς απόδοσης, καλύτερη είναι η QAM64 και από πλευράς αντοχής σε δυσχερείς τηλεπικοινωνιακές συνθήκες, καλύτερη είναι η QAM4. Ο symbol mapper δέχεται τα υπο διαμόρφωση k-blocks στην είσοδο *in_data* και κατόπιν - αφού αφαιρέσει τα τυχόντα bit πλεονασμού - τα αντιστοιχεί σε QAM σύμβολα σύμφωνα με τις επιταγές του σήματος *sym_block_size*, το οποίο περιέχει τον επιθυμητό τύπο διαμόρφωσης για το τρέχον block. Έπειτα από ένα αρχικό delay 1 κύκλου ρολογιού, η ροή των διαμορφωμένων κατά QAM συμβόλων είναι συνεχής.

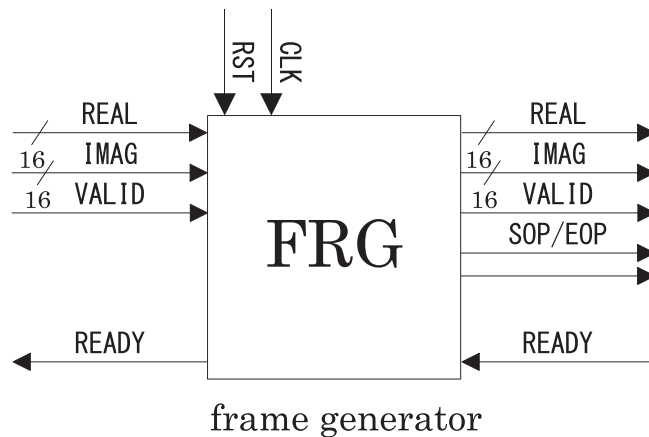
Η έξοδος δεδομένων δεν είναι πια μοναδική αλλά, αντίθετα, υπάρχουν δυο έξοδοι δεδομένων, μία για το πραγματικό και μία για το φανταστικό μέρος του μιγαδικού -πλέον- σήματος πληροφορίας (*out_real* και *out_imag*, αντίστοιχα).

Από συστημικής απόψεως, ο symbol mapper δεν εισάγει φουσαλλίδες στο datapath και δεν ασκεί frontpressure στα downstream modules, ούτε backpressure στα upstream.

Frame Generator

Στο Σχήμα 3.18 παρουσιάζεται το διάγραμμα ακροδεκτών του Frame Generator και ακολουθεί μια συνοπτική περιγραφή του κάθε ακροδέκτη.

```
entity frame_gen is
  generic (
    DIN_WIDTH : natural;
    -- OFDM subcarrier count
    N_SC      : natural;
    -- training symbol count per OFDM frame
    N_TRAIN  : natural;
    -- data symbol count per OFDM frame
    N_DATA   : natural
  );
```

Σχήμα 3.18: Διάγραμμα ακροδεκτών Frame Generator

```

port (
  clk      : in std_logic;
  rst_n    : in std_logic;
  -- Input Avalon
  in_real  : in std_logic_vector(DIN_WIDTH-1 downto 0);
  in_imag  : in std_logic_vector(DIN_WIDTH-1 downto 0);
  in_valid : in std_logic;
  in_ready : out std_logic;
  -- Output Avalon
  out_real : out std_logic_vector(DIN_WIDTH-1 downto 0);
  out_imag : out std_logic_vector(DIN_WIDTH-1 downto 0);
  out_valid : out std_logic;
  out_sop  : out std_logic;
  out_eop  : out std_logic;
  out_ready : in std_logic
);
end frame_gen;

```

Ακροδέκτες Εισόδου

clk ▷ Το βασικό ρολόι του συστήματος.

rst_n ▷ Το βασικό σήμα επαναφοράς του συστήματος, αρνητικής λογικής.

in_real ▷ Είσοδος δεδομένων εύρους 16 bit. Περιέχει την πραγματική συνιστώσα του μιγαδικού σήματος που προκύπτει μετά τη διαμόρφωση κατά QAM. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_imag ▷ Είσοδος δεδομένων εύρους 16 bit. Περιέχει τη φανταστική συνιστώσα του μιγαδικού σήματος που προκύπτει μετά τη διαμόρφωση κατά QAM. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_valid ▷ Είσοδος που διεκπεραιώνει το ρόλο *sink valid* του Avalon ST Interface.

out_ready ▷ Είσοδος που διεκπεραιώνει το ρόλο *source ready* του Avalon ST Interface.

Ακροδέκτες Εξόδου

out_real ▷ Έξοδος δεδομένων εύρους 16 bit. Περιέχει το πραγματικό μέρος του μιγαδικού σήματος πληροφορίας. Ανά περίπτωση αυτό το σήμα είναι είτε το διαμορφωμένο κατά QAM X (X=4|16|64) δεδομένο του προηγούμενου σταδίου, είτε κάποιο, διαμορφωμένο κατά QAM 4, κομμάτι ενός training symbol. Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_imag ▷ Έξοδος δεδομένων εύρους 16 bit. Περιέχει το φανταστικό μέρος του μιγαδικού σήματος πληροφορίας. Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_valid ▷ Έξοδος που διεκπεραιώνει το ρόλο *source valid* του Avalon ST Interface.

out_sop ▷ Έξοδος που διεκπεραιώνει το ρόλο *startofpacket* του Avalon ST Interface. Σηματοδοτεί την αρχή ενός OFDM συμβόλου στη συχνότητα.

out_eop ▷ Έξοδος που διεκπεραιώνει το ρόλο *endofpacket* του Avalon ST Interface. Σηματοδοτεί το τέλος ενός OFDM συμβόλου στη συχνότητα

in_ready ▷ Έξοδος που διεκπεραιώνει το ρόλο *sink ready* του Avalon ST Interface.

Η πιο βασική εργασία που επιτελεί ο frame generator είναι η παρεμβολή στα σύμβολα πληροφορίας (data symbols) των δύο - ανά frame - training συμβόλων. Τα δύο πρώτα χρησιμοποιούνται από τον αλγόριθμο Schmidl-Cox για την επίτευξη συγχρονισμού στο χρόνο (στο δέκτη, στο module *time sync*) και το τρίτο από τον αλγόριθμο LS για την εκτίμηση καναλιού (στο δέκτη, στο module *channel estimator*). Έτσι, μετά από προκαθορισμένο αριθμό data symbols που έχουν περάσει από τον frame generator (χωρίς να υποστούν επεξεργασία), η ροή διακόπτεται και παρεμβάλλονται τα training symbols. Προκειμένου να επιτευχθεί αυτό, ο frame generator είναι ενήμερος για τον αριθμό των data συμβόλων που έχουν διέλθει από αυτόν, κάθε φορά.

Παράλληλα, ο frame generator επιτελεί και μια ομαδοποίηση των διαμορφωμένων κατά QAM συμβόλων (ή αλλιώς *sub-symbols*) δημιουργώντας το ωφέλιμο φορτίο (payload) ενός OFDM συμβόλου. Από αυτό το module και πέρα αποκτά υπόσταση η έννοια του "πακέτου" (ipacket) με αυτή να αντιστοιχεί στο OFDM σύμβολο. Σχετικές με αυτή είναι οι θύρες εξόδου *out_sop* και *out_eop* που σηματοδοτούν την αρχή και το τέλος του πακέτου, αντίστοιχα. Ταυτόχρονα, παράγει το OFDM frame σαν ομαδοποίηση των OFDM συμβόλων. Είναι ακριβώς η αρχή κάθενός OFDM frame το μέρος στο οποίο παρεμβάλλονται τα training σύμβολα, που χρησιμοποιούνται στο δέκτη, για τη σωστή αποδιαμόρφωση του σήματος.

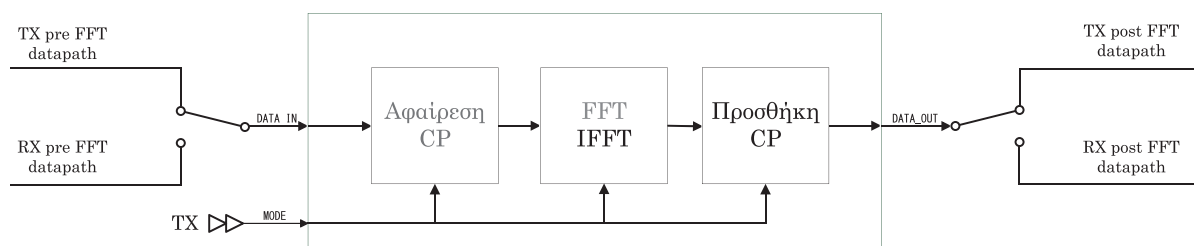
Από συστημικής απόψεως, τώρα, ακριβώς λόγω της διαδικασίας παρεμβολής των training συμβόλων, ο frame generator ασκεί backpressure στα modules του upstream. Αντίθετα, σε κανονικές συνθήκες λειτουργίας δεν προβλέπεται χρήση του μηχανισμού frontpressure από αυτό το module.

Παράμετρος	Σύμβολο	Περιγραφή
N_{SC}	N	Αριθμός subcarrier και μέγεθος FFT
N_{TRAIN}	N_{train}	Πλήθος Training Συμβόλων ανά OFDM Frame
N_{DATA}	$R - N_{train}$	Πλήθος Data Συμβόλων ανά OFDM Frame

Πίνακας 3.9: Παράμετροι του frame generator

Altera OFDM Core • Modulator

Στο Σχήμα 3.19 παρουσιάζεται εποπτικά ο τρόπος λειτουργίας του module κατά τη λειτουργία πομπού. Οι υπομονάδες που δε χρησιμοποιούνται εμφανίζονται γραμμοσκιασμένες.



Σχήμα 3.19: Εποπτεία λειτουργίας του module OFDM Core (TX Mode)

Όπως αναφέρθηκε αυτό το module χρησιμοποιείται, τόσο κατά την εκπομπή, όσο και κατά τη λήψη. Εδώ θα παρουσιαστεί η λειτουργία του ως μέρος της αλυσίδας του πομπού.

Οι λειτουργίες που λαμβάνουν χώρα εδώ, είναι η διαμόρφωση κατά OFDM και η προσθήκη της κυκλικής επέκτασης. Τα baseband δεδομένα (ωφέλιμα OFDM σύμβολα στη συχνότητα) τροφοδοτούνται απευθείας (το CP-removal submodule είναι διάφανο ως προς τα δεδομένα εισόδου στη λειτουργία πομπού) στο IFFT module.²⁶ Η έξοδος του IFFT οδηγείται στο *CP-Insertion* submodule όπου και λαμβάνει χώρα η εισαγωγή του κυκλικού προθέματος. Μετά το πέρας αυτού του σταδίου, τα πλήρη OFDM σύμβολα (στο χρόνο) είναι έτοιμα προς μετάδοση.

Τα isymbol στις εισόδους *sink_real*, *sink_imag*, αποτελούνται από (μιγαδικά) subsymbols, διαμορφωμένα κατά QAM. Τα iracket όπως ορίζονται από τα *sink_sop*, *sink_eop* αποτελούνται από ωφέλιμα OFDM σύμβολα στη συχνότητα. Τα isymbol, στις εξόδους *source_real*, *source_imag*, αποτελούνται από μιγαδικά samples του σήματος προς μετάδοση στο χρόνο, ενώ τα iracket, όπως ορίζονται από τα *source_sop*, *source_eop*, αποτελούνται από OFDM σύμβολα. Ωστόσο η έννοια του πακέτου σταματά να υφίσταται μετά από αυτό το module, καθώς στον DAC αυτή δεν έχει σημασία.

Απο συστημική άποψη, το module αυτό δεν εισάγει frontpressure στην downstream αλυσίδα (όπως, άλλωστε, συμβαίνει σε όλα τα modules της TX αλυσίδας). Ωστόσο, ασκεί το μηχανισμό backpressure στο upstream λόγω της διαδικασίας εισαγωγής του κυκλικού προθέματος στα σύμβολα OFDM.

3.5 Αλυσίδα Υποσυστήματος Δέκτη

3.5.1 Η Διαδρομή Δεδομένων στο Δέκτη (RX Datapath)

Στο σχήμα 3.20 παρουσιάζεται η πλήρης αλυσίδα του δέκτη. Τα κύρια modules που τη συναποτελούν είναι τα :

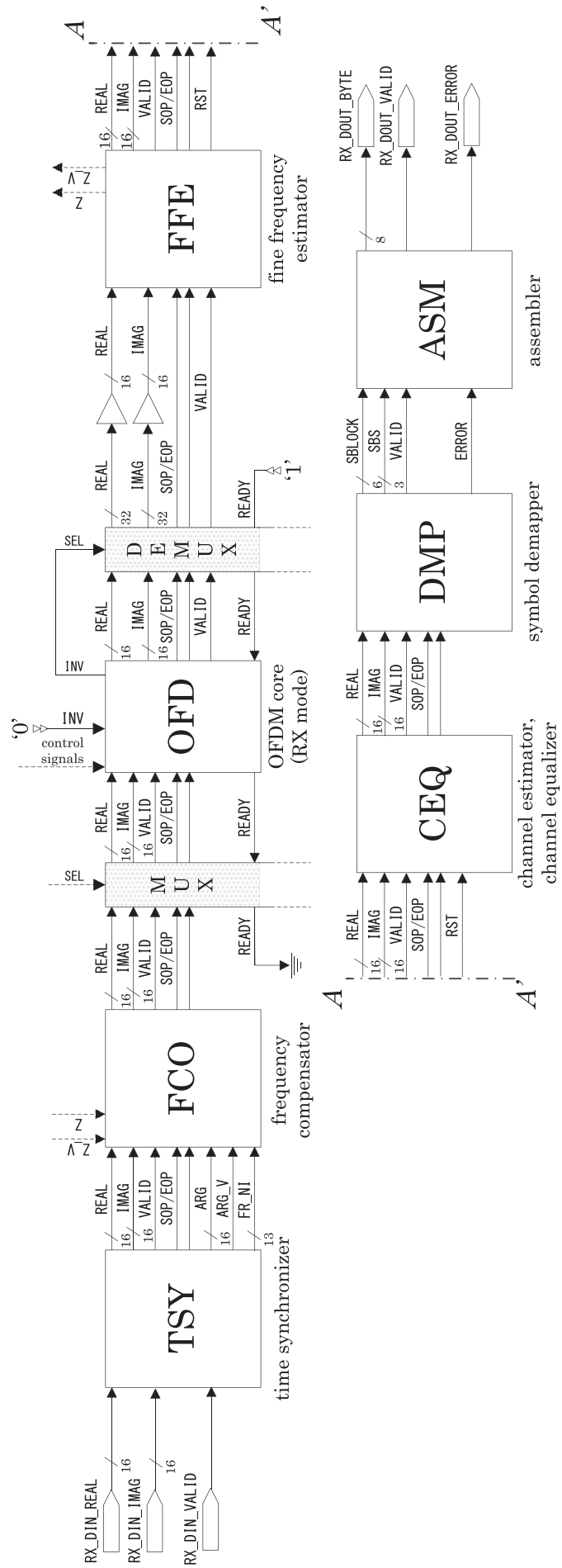
- *Time Synchronizer* ή Συγχρονιστής στο χρόνο
- *Frequency Compensator* ή Αντισταθμιστής συχνότητας
- *OFDM Demodulator* ή Αποδιαμορφωτής κατά OFDM
- *Fine Frequency Estimator* ή Ραφινάτος Εκτιμητής Συχνότητας
- *Channel Estimator* ή Εκτιμητής Καναλιού
- *Symbol Demapper* ή Αποδιαμορφωτής κατά QAM
- *Assembler* ή Συγκεραστής

Συνοπτική Περιγραφή της Πορείας της Πληροφορίας στο Δέκτη

Παραδοχές Τα δεδομένα πληροφορίας βρίσκονται σε ψηφιακή μορφή. Μπορεί να αποτελούν κάποιο αρχείο στον υπολογιστή ή μέσω ADC να έχουν περάσει στο σύστημα. Η βασική υπόθεση, εδώ, είναι οτι -τα διαμορφωμένα κατά OFDM δεδομένα- έχουν περάσει από κάποιο κανάλι. Υπενθυμίζεται ότι το σύστημα είναι ικανό να αποδιαμορφώσει δεδομένα που έχουν περάσει από κανάλι τύπου :

- Απλό σύρμα (wire), δηλαδή ενσύρματο κανάλι στο οποίο δεν υπάρχει AWGN
- Ασύρματο κανάλι (wireless), παρουσία θορύβου AWGN μόνο

²⁶Το CP-removal submodule δεν εκτελεί κάποια επεξεργασία στο σήμα. Ωστόσο περιέχει την απαραίτητη λογική για τη διάσχιση των clock domain. Περισσότερα για αυτό, στο Κεφάλαιο 4.



Σχήμα 3.20: Το RX Datapath

- Ασύρματο συχνοεπιλεκτικό κανάλι (wireless frequency-selective channel), παρουσία θορύβου AWGN και slow-fading multipath

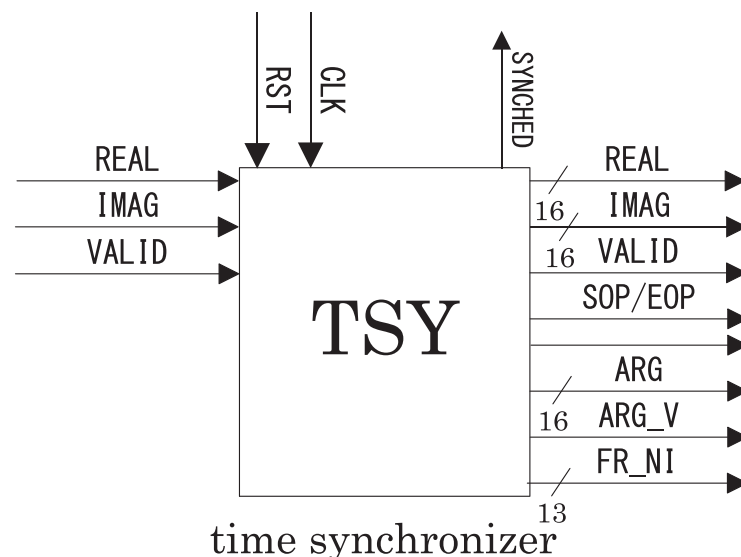
ενώ η επεξεργασία των δεδομένων γίνεται στο baseband. Μία πιθανή προϊστορία των δεδομένων περιλαμβάνει τη λήψη τους από κεραία, τον υποβιβασμό τους από τις RF στις IF συχνότητες, την αποδιαμόρφωσή τους σε μιγαδικό σήμα βασικής ζώνης, αποτελούμενο από τις συνιστώσες I,Q και την ψηφιοποίησή τους μέσω 2 ADC.

Τώρα, το πρώτο -και πιο σημαντικό- στάδιο κατά την είσοδο των δεδομένων στο σύστημα είναι ο συγχρονισμός στο χρόνο, δηλαδή η χρονική ταυτοποίηση της αρχής του frame των λαμβανόμενων δεδομένων. Αυτή είναι η κύρια δουλειά του module *time sync*, με δεύτερη την coarse εκτίμηση συχνότητας.. Μετά την έξοδο των δεδομένων από αυτό το module, ακολουθεί η διαδικασία της αντιστάθμισης συχνότητας, κάτι το οποίο επιτελείται στο module *frequency compensator*. Ακολουθεί η αφαίρεση του κυκλικού προθέματος και η εφαρμογή του μετασχηματισμού FFT στα δεδομένα. Με αυτό τον τρόπο ολοκληρώνεται η αποδιαμόρφωση κατά OFDM των δεδομένων. Αυτό λαμβάνει χώρα στο module *OFDM_Demodulator* (Altera). Σε ενδεχόμενη μελλοντική επέκταση το επόμενο module, δηλαδή το *fine frequency estimator* θα επιτελεί την fine εκτίμηση συχνότητας. Προς το παρόν αυτό το module αφήνει τα δεδομένα ανεπερέργαστα και η μόνη λειτουργία του είναι η αφαίρεση των δύο πρώτων training symbol. Επόμενο στάδιο είναι η εκτίμηση καναλιού και η ισοστάθμιση των δεδομένων. Σε αυτό το στάδιο έχουν απαλειφθεί οι παρενέργειες που ήταν παρούσες στο σήμα λόγω της μετάδοσης και της διέλευσης από το τηλεπικοινωνιακό κανάλι. Ακολουθεί η αποδιαμόρφωση κατά QAM, κάτι που λαμβάνει χώρα στο module *symbol demapper*. Τελικό στάδιο πριν την έξοδο των δεδομένων από το σύστημα είναι η "συναρμολόγηση" των κομματιών πληροφορίας (k-blocks) σε bytes και αυτό ακριβώς είναι που επιτελεί ο *assembler*.

3.5.2 Συστημική Περιγραφή Μονάδων

Time Synchronizer

Στο Σχήμα 3.21 παρουσιάζεται το διάγραμμα ακροδεκτών του Time Synchronizer και ακολουθεί μια συνοπτική περιγραφή του κάθε ακροδέκτη.



Σχήμα 3.21: Διάγραμμα ακροδεκτών Time Synchronizer

```
entity time_sync is
  generic (
```

```

DIN_WIDTH          : natural;
-- sample frame counter width ( must be >= log2(SZ_SYM*SZ_FRAME) )
NI_FR_WIDTH        : natural;
-- cyclic prefix size
SZ_CP              : natural;
-- useful symbol duration --MAX 1024
SZ_USYM           : natural;
-- log2(SZ_USYM)
LOG2_SZ_USYM      : natural;
-- frame size (in symbols)
SZ_FRAME          : natural;
-- Metric valid (not noise or data) threshold as percentage of the max value (theory max val
=1.0)
-- Trial and error parameter (per chan)
METRIC_VALID_THRES_PCNT : natural;
-- Metric Observation window size in samples (Z) TYP:0.5*SZ_USYM; MAX: 1*SZ_USYM=1024
METRIC_SEARCH_SPAN  : natural;
SOF_OFFSET         :natural -- always <= SZ_CP
);
port (
  clk      : in std_logic;
  rst      : in std_logic;
  in_real  : in std_logic_vector(DIN_WIDTH-1 downto 0);
  in_imag  : in std_logic_vector(DIN_WIDTH-1 downto 0);
  in_valid : in std_logic;
  out_synced : out std_logic;
  out_fr_ni : out std_logic_vector(NI_FR_WIDTH-1 downto 0); -- unsigned
  out_real  : out std_logic_vector(DIN_WIDTH-1 downto 0);
  out_imag  : out std_logic_vector(DIN_WIDTH-1 downto 0);
  out_arg   : out std_logic_vector(15 downto 0); --signed FP <s/16/13>
  out_arg_v : out std_logic;
  out_valid : out std_logic;
  out_sop   : out std_logic;
  out_eop   : out std_logic
);
end time_sync ;

```

Ακροδέκτες Εισόδου

clk ▷ Το βασικό ρολόι του συστήματος.

rst ▷ Το βασικό σήμα επαναφοράς του συστήματος.

in_real ▷ Είσοδος δεδομένων, εύρους 16 bit. Πρόκειται για το πραγματικό μέρος του μιγαδικού σήματος OFDM, έτσι όπως αυτό έχει προκύψει από τον ADC ή από κάποια ψηφιακή πηγή. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_imag ▷ Είσοδος δεδομένων, εύρους 16 bit. Πρόκειται για το φανταστικό μέρος του μιγαδικού σήματος OFDM, έτσι όπως αυτό έχει προκύψει από τον ADC ή από κάποια ψηφιακή πηγή. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_valid ▷ Είσοδος που διεκπεραιώνει το ρόλο *sink valid* του Avalon ST Interface.

Ακροδέκτες Εξόδου

out_synced ▷ Έξοδος εύρους 1 bit, υποδηλώνει την επίτευξη (on) ή όχι (off) του συγχρονισμού στο χρόνο. Μπορεί να οδηγήσει κατάλληλο ενδείκτη LED.

out_fr_ni ▷ Έξοδος δείκτης του A/A του τρέχοντος sample. Χρησιμοποιείται από τον frequency compensator.

out_real ▷ Έξοδος δεδομένων εύρους 16 bit. Αντιστοιχεί στο πραγματικό μέρος του μιγαδικού σήματος. Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_imag ▷ Έξοδος δεδομένων εύρους 16 bit. Αντιστοιχεί στο φανταστικό μέρος του μιγαδικού σήματος.

Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_arg ▷ Έξοδος που περιέχει τη γωνία στροφής των δεδομένων (σε °) λόγω ολίσθησης συχνότητας.

out_arg_v ▷ Έξοδος που σηματοδοτεί την εγκυρότητα (ή μη) των περιεχομένων της εξόδου *out_arg*

out_valid ▷ Έξοδος που διεκπεραιώνει το ρόλο *source valid* του Avalon ST Interface.

out_sop ▷ Έξοδος που διεκπεραιώνει το ρόλο *startofpacket* του Avalon ST Interface.

out_eop ▷ Έξοδος που διεκπεραιώνει το ρόλο *endofpacket* του Avalon ST Interface.

Αντικείμενο του *time sync* είναι η εκτέλεση της πιο βασικής εργασίας κατά τη λήψη της μεταδιδόμενης πληροφορίας από το δέκτη: ο συγχρονισμός στο χρόνο και η πρώτη εκτίμηση της ολίσθησης συχνότητας. Μετά την είσοδο των δεδομένων από τις θύρες *in_real* και *in_imag*, εκτελείται στο εσωτερικό του module ο αλγόριθμος Schmidl-Cox ο οποίος έχει σαν "πρώτη ύλη" το πρώτο training σύμβολο του ληφθέντος OFDM frame. Αποτέλεσμα είναι η σηματοδότηση της αρχής των λαμβανόμενων δεδομένων δηλαδή η αρχή του OFDM frame. Από αυτό το σημείο της αλυσίδας, έχει νόημα το *ipacket*, με το καθένα από αυτά να αντιστοιχεί σε ένα OFDM σύμβολο. Η αρχή και το πέρας του *ipacket* προσδιορίζονται από τα σήματα *sop* και *eop*, που παράγει ο *time synchronizer*. Σε αυτό το σημείο, αν και έχει παρέλθει η χρησιμότητα του πρώτου training συμβόλου αυτό δεν αποπέμπεται από το frame, καθώς θα το χρειαστεί ένα module του downstream, ο *fine frequency estimator*.

Τώρα, παράλληλα με το συγχρονισμό του δέκτη και χρησιμοποιώντας το ίδιο training σύμβολο, το module αυτό διενεργεί και μία εκτίμηση της γωνίας της στροφής που επιβάλλει το φαινόμενο της ολίσθησης συχνότητας. Η εκτίμηση αυτή παρέχεται στην έξοδο *out_arg* και αφορά στα samples όλου του frame που ακολουθούν, χρησιμοποιείται δε, από το επόμενο module στην αλυσίδα του δέκτη, τον *frequency compensator*.

Η εγκυρότητα της τιμής της σηματοδοτείται από το *arg_v*. Είναι προφανές ότι το module αυτό έχει δύο λογικές εξόδους: η κύρια έξοδος φέρει τα samples του σήματος συγχρονισμένα στο χρόνο, ενώ η δευτερεύουσα φέρει την εκτίμηση φάσης, η οποία είναι διαθέσιμη για έναν κύκλο, στην αρχή του κάθε frame. Αντιθέτως τα δεδομένα της κύριας εξόδου μεταφέρονται σε κάθε κύκλο από τη στιγμή που βρεθεί η αρχή του frame.

Από συστημική άποψη, ο *time sync* δεν επιβάλλει, φυσικά, *backpressure*. Επιβάλλει, όμως, έμπροσθεν πέδηση (*frontpressure*) στα module του downstream, για όσο χρόνο απαιτείται προκειμένου να ολοκληρωθεί η διαδικασία του συγχρονισμού και η εκτίμηση της ολίσθησης συχνότητας.

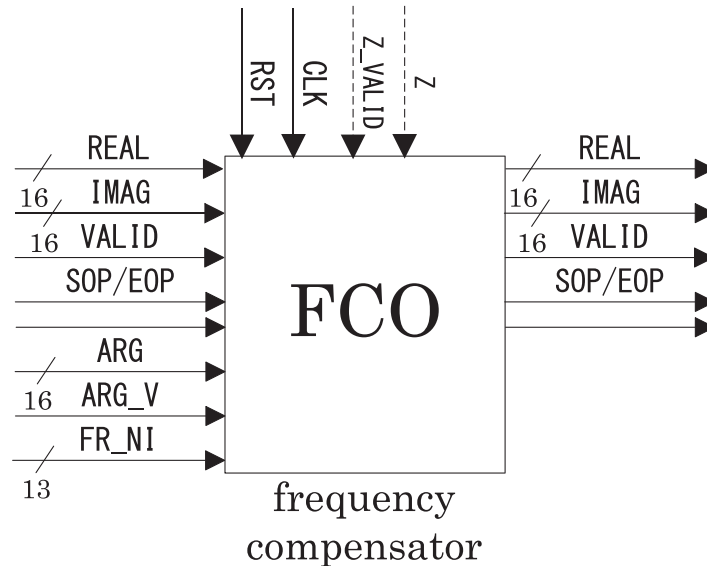
Παράμετρος	Σύμβολο	Περιγραφή
<i>SZ_CP</i>	<i>N_{CP}</i>	Μήκος κυκλικού προθέματος σε samples
<i>SZ_FRAME</i>	<i>R</i>	Μήκος frame σε σύμβολα

Πίνακας 3.10: Παράμετροι του *time sync*

Frequency Compensator

Στο Σχήμα 3.22 παρουσιάζεται το διάγραμμα ακροδεκτών του Frequency Compensator και ακολουθεί μια συνοπτική περιγραφή του κάθε ακροδέκτη.

```
entity freq_compensator is
  generic(
    DIN_WIDTH : natural;
    LOG2_SZ_FFT : natural
  );
  port(
    clk,rst : in std_logic;
    in_phi  : in std_logic_vector(15 downto 0); --signed
    in_n    : in std_logic_vector(12 downto 0);
```



Σχήμα 3.22: Διάγραμμα ακροδεκτών Frequency Compensator

```

in_phi_v : in std_logic;
in_z      : in std_logic_vector(10 downto 0); --max timh 2 be trimmed
in_z_valid : in std_logic;
-----Avalon-----
--Input
in_real   : in std_logic_vector(DIN_WIDTH-1 downto 0);
in_imag   : in std_logic_vector(DIN_WIDTH-1 downto 0);
in_sop    : in std_logic;
in_eop    : in std_logic;
in_valid  : in std_logic;
--Output
out_real  : out std_logic_vector(DIN_WIDTH-1 downto 0);
out_imag  : out std_logic_vector(DIN_WIDTH-1 downto 0);
out_sop   : out std_logic;
out_eop   : out std_logic;
out_valid : out std_logic
);
end freq_compensator;

```

Ακροδέκτες Εισόδου

clk ▷ Το βασικό ρολόι του συστήματος.

rst ▷ Το βασικό σήμα επαναφοράς του συστήματος.

in_n ▷ Είσοδος δείκτης του A/A του τρέχοντος sample.

in_phi ▷ Είσοδος που περιέχει τη γωνία στροφής των δεδομένων (σε °) λόγω ολίσθησης συχνότητας.

in_phi_v ▷ Είσοδος που αναφέρεται στην εγκυρότητα (ή μη) των περιεχομένων της εισόδου *in_arg*

in_z ▷ Ακέραιο μέρος της γωνίας στροφής φάσης. Σε μελλοντική επέκταση του συστήματος θα είναι η είσοδος ανάδρασης από τη διαδικασία εκτίμησης συχνότητας (fine frequency estimation). Στην παρούσα υλοποίηση η τιμή της είναι αδιάφορη (υπονοείται εσωτερικά ότι έχει τιμή 0).

in_z_valid ▷ Μελλοντικά, θα αναφέρεται στην εγκυρότητα (ή μη) της εισόδου *in_z*. Στην παρούσα υλοποίηση είναι αδιάφορη.

in_real ▷ Είσοδος δεδομένων 16 bit. Αντιστοιχεί στο πραγματικό μέρος του μιγαδικού σήματος. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_imag ▷ Είσοδος δεδομένων 16 bit. Αντιστοιχεί στο φανταστικό μέρος του μιγαδικού σήματος. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_sop ▷ Είσοδος που διεκπεραιώνει το ρόλο *startofpacket* του Avalon ST Interface.

in_eop ▷ Είσοδος που διεκπεραιώνει το ρόλο *endofpacket* του Avalon ST Interface.

in_valid ▷ Είσοδος που διεκπεραιώνει το ρόλο *sink valid* του Avalon ST Interface.

Ακροδέκτες Εξόδου

out_real ▷ Έξοδος δεδομένων εύρους 16 bit. Περιέχει το πραγματικό μέρος του - αντισταθμισμένου στη συχνότητα - μιγαδικού σήματος. Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_imag ▷ Έξοδος δεδομένων εύρους 16 bit. Περιέχει το φανταστικό μέρος του - αντισταθμισμένου στη συχνότητα - μιγαδικού σήματος. Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_valid ▷ Έξοδος που διεκπεραιώνει το ρόλο *source valid* του Avalon ST Interface.

out_sop ▷ Έξοδος που διεκπεραιώνει το ρόλο *startofpacket* του Avalon ST Interface.

out_eop ▷ Έξοδος που διεκπεραιώνει το ρόλο *endofpacket* του Avalon ST Interface.

Σε αυτό το module είναι που επιτελείται η αντιστάθμιση συχνότητας. Η εκτίμηση ολίσθησης συχνότητας που έλαβε χώρα στον *time sync* εισέρχεται από τη θύρα *in_phi* και χρησιμοποιείται για την αντιστάθμιση του σήματος, σύμφωνα με το μαθηματικό τύπο

$$r_c(t) = r(t) e^{-j2\pi t \hat{D}_f} = r(t) e^{-j \frac{2t\hat{\phi}}{T_o}}$$

όπου $\hat{\phi}$ είναι η εκτίμηση. Το αντισταθμισμένο μιγαδικό σήμα λαμβάνεται (πραγματικό και φανταστικό μέρος) από τις εξόδους *out_real* και *out_imag*.

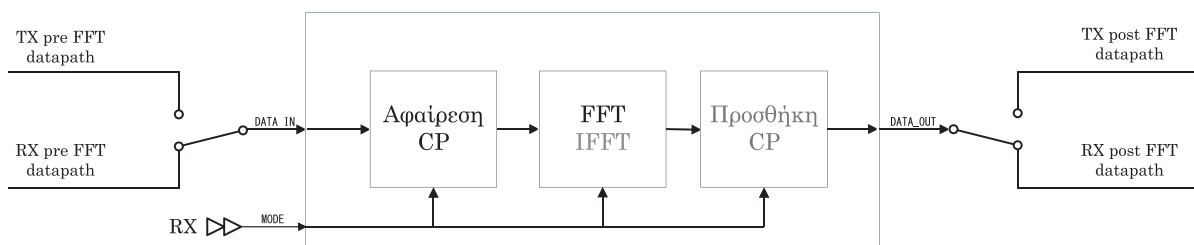
Θεωρώντας τη συστημική συμπεριφορά του εν λόγω module, δεν ασκείται από αυτό backpressure αλλά ούτε και frontpressure. Η ροή των δεδομένων είναι συνεχής μετά από αρχική καθυστέρηση ?? κύκλων ρολογιού.

Παράμετρος	Σύμβολο	Περιγραφή
<i>LOG2_SZ_FFT</i>	$\log_2 N$	Διαδικός λογάριθμος του μεγέθους του FFT

Πίνακας 3.11: Παράμετροι του frequency compensator

Altera OFDM Core • Demodulator

Στο Σχήμα 3.23 παρουσιάζεται εποπτικά ο τρόπος λειτουργίας του module κατά τη λειτουργία δέκτη. Οι υπομονάδες που δε χρησιμοποιούνται εμφανίζονται γραμμοσκιασμένες.



Σχήμα 3.23: Εποπτεία λειτουργίας του module OFDM Core (RX Mode)

Όπως έχει αναφερθεί, αυτό το module χρησιμοποιείται, τόσο κατά την εκπομπή, όσο και κατά τη λήψη. Εδώ θα επεξηγηθεί η λειτουργία του ως μέρος της αλυσίδας του δέκτη.

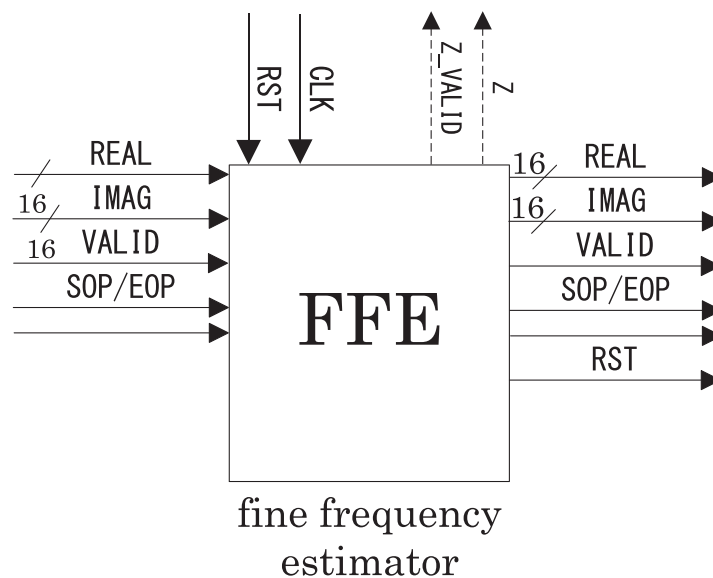
Εδώ είναι που λαμβάνει χώρα η καθαυτό αποδιαμόρφωση κατά OFDM. Το πρώτο στάδιο είναι η αφαίρεση του κυκλικού προθέματος από τα εισερχόμενα OFDM σύμβολα. Το *CP-Removal module* αναζητά τη σωστή αρχή ενός OFDM συμβόλου και τροφοδοτεί τα δεδομένα στο FFT module για αποδιαμόρφωση. Στη συνέχεια τα δεδομένα περνούν μέσω του CP-addition module στην έξοδο, χωρίς περαιτέρω επεξεργασία.

Τα isymbol στις εισόδους *sink_real*, *sink_imag*, αποτελούνται από μιγαδικά samples του σήματος (στο πεδίο του χρόνου). Τα ipacket όπως ορίζονται από τα *sink_sop*, *sink_eop* αποτελούνται από OFDM σύμβολα, όπως έχουν ανιχνευτεί από τη διαδικασία του συγχρονισμού. Στις εξόδους *source_real*, *source_imag*, τα isymbol, αποτελούνται από subsymbol, ενώ τα ipacket, όπως ορίζονται από τα *source_sop*, *source_eop*, αποτελούνται από ωφέλιμα OFDM σύμβολα στη συχνότητα.

Απο συστημική άποψη, το module αυτό δεν εισάγει backpressure στην upstream αλυσίδα (όπως, άλλωστε, συμβαίνει σε όλα τα modules της RX αλυσίδας). Ωστόσο, ασκεί το μηχανισμό έμπροσθεν πέδησης (frontpressure) στο downstream λόγω της διαδικασίας αφαίρεσης του κυκλικού προθέματος από τα OFDM σύμβολα.

Fine Frequency Estimator

Στο Σχήμα 3.24 παρουσιάζεται το διάγραμμα ακροδεκτών του Fine Frequency Estimator και ακολουθεί μια συνοπτική περιγραφή του κάθε ακροδέκτη.



Σχήμα 3.24: Διάγραμμα ακροδεκτών Fine Frequency Estimator

```
entity fine_freq_estimator is
  generic(
    DIN_WIDTH : natural;
    Z_WIDTH   : natural; -- log2searchspan
    SZ_FRAME  : natural
  );
  port(
    clk,rst   : in std_logic;
    -----Avalon-----
    --Input
    in_real   : in std_logic_vector(DIN_WIDTH-1 downto 0);
    in_imag   : in std_logic_vector(DIN_WIDTH-1 downto 0);
    in_sop    : in std_logic;
    in_eop    : in std_logic;
    in_valid  : in std_logic;
```

```

--Output
out_real  : out std_logic_vector(DIN_WIDTH-1 downto 0);
out_imag  : out std_logic_vector(DIN_WIDTH-1 downto 0);
out_rst   : out std_logic;
out_sop   : out std_logic;
out_eop   : out std_logic;
out_valid : out std_logic;
-----Other-----
out_z     : out std_logic_vector(Z_WIDTH-1 downto 0);
out_z_valid : out std_logic
);
end fine_freq_estimator;

```

Ακροδέκτες Εισόδου

clk ▷ Το βασικό ρολόι του συστήματος.

rst ▷ Το βασικό σήμα επαναφοράς του συστήματος.

in_real ▷ Είσοδος εύρους 16 bit. Περιέχει το πραγματικό μέρος του μιγαδικού σήματος. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_imag ▷ Είσοδος εύρους 16 bit. Περιέχει το φανταστικό μέρος του μιγαδικού σήματος. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_sop ▷ Είσοδος που διεκπεραιώνει το ρόλο *startofpacket* του Avalon ST Interface.

in_eop ▷ Είσοδος που διεκπεραιώνει το ρόλο *endofpacket* του Avalon ST Interface.

in_valid ▷ Είσοδος που διεκπεραιώνει το ρόλο *sink valid* του Avalon ST Interface.

Ακροδέκτες Εξόδου

out_real ▷ Έξοδος δεδομένων εύρους 16 bit. Περιέχει το πραγματικό μέρος του μιγαδικού σήματος. Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_imag ▷ Έξοδος δεδομένων εύρους 16 bit. Περιέχει το φανταστικό μέρος του μιγαδικού σήματος. Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_valid ▷ Έξοδος που διεκπεραιώνει το ρόλο *source valid* του Avalon ST Interface.

out_sop ▷ Έξοδος που διεκπεραιώνει το ρόλο *startofpacket* του Avalon ST Interface.

out_eop ▷ Έξοδος που διεκπεραιώνει το ρόλο *endofpacket* του Avalon ST Interface.

out_rst ▷ Έξοδος του σήματος ελέγχου *out_rst*. Σηματοδοτεί το τελευταίο data symbol του frame.

out_z ▷ Ακέραιο μέρος της εκτίμησης στροφής φάσης, λόγω ολίσθησης συχνότητας. Σε μελλοντική επέκταση του συστήματος θα είναι η έξοδος ανάδρασης προς τη διαδικασία αντιστάθμισης συχνότητας (frequency compensation). Στην παρούσα υλοποίηση έχει σταθερά την τιμή '0' και η θύρα δε συνδέεται πουθενά.

out_z_valid ▷ Μελλοντικά, θα αναφέρεται στην εγκυρότητα (ή μη) της εξόδου *in_z*. Στην παρούσα υλοποίηση είναι πάντα ενεργοποιημένη.

Στην §2.2.4 προδιαγράφεται το υποσύστημα *Fine Frequency Estimator* το οποίο δεν κρίθηκε υλοποιήσιμο στο παρόν σύστημα, λόγω κάλυψης των προδιαγραφών από το coarse frequency estimation, που γίνεται στον time synchronizer. Επειδή, όμως, μπορεί κάλλιστα να αποτελέσει μια μελλοντική επέκταση του συστήματος, κρίθηκε σωστή η τοποθέτηση ενός stub module στην αλυσίδα προκειμένου να καταστεί ευκολότερη μια ενδεχόμενη αναβάθμιση. Το stub module αυτό, πληροί τις προδιαγραφές και τις απαιτήσεις του Avalon ST Interface και η (μελλοντική) λειτουργία του είναι η παροχή (εν είδει ανάδρασης) του offset συχνότητας μέσω της θύρας εξόδου *out_z*, στον *frequency compensator*. Η κύρια λειτουργία του, στην τωρινή του μορφή είναι η αφαίρεση από τα frame των 2 πρώτων training συμβόλων.

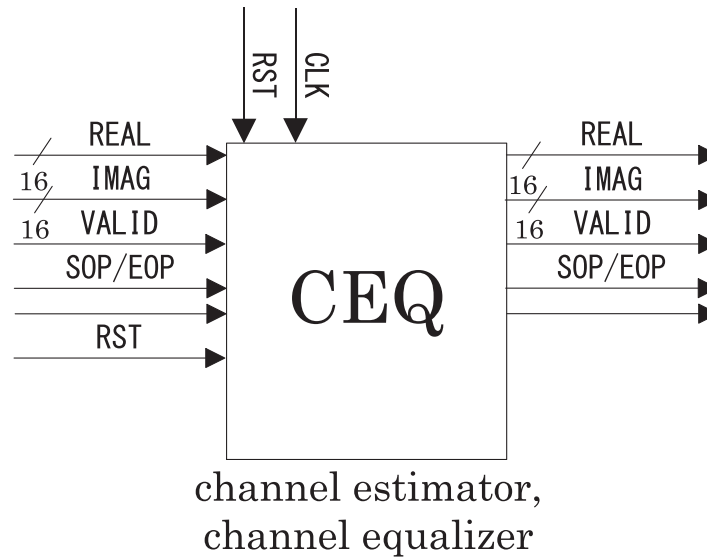
Από συστημική άποψη, ο *fine frequency estimator* ασκεί το μηχανισμό frontpressure στο down-stream λόγω της διαδικασίας αφαίρεσης των training συμβόλων.

Παράμετρος	Σύμβολο	Περιγραφή
<i>SZ_FRAME</i>	<i>R</i>	Μήκος frame σε σύμβολα

Πίνακας 3.12: Παράμετροι του fine frequency estimator

Channel Estimator

Στο Σχήμα 3.25 παρουσιάζεται το διάγραμμα ακροδεκτών του Channel Estimator και ακολουθεί μια συνοπτική περιγραφή του κάθε ακροδέκτη.



Σχήμα 3.25: Διάγραμμα ακροδεκτών Channel Estimator

```
entity channel_estimator is
  generic(
    DIN_WIDTH : natural;
    FFT_SIZE : natural
  );
  port(
    clk,rst : in std_logic;
    in_rst : in std_logic;
    -----Avalon-----
    --Input
    in_real : in std_logic_vector(DIN_WIDTH-1 downto 0);
    in_imag : in std_logic_vector(DIN_WIDTH-1 downto 0);
    in_sop : in std_logic;
    in_eop : in std_logic;
    in_valid : in std_logic;
    --Output
    out_real : out std_logic_vector(DIN_WIDTH-1 downto 0);
    out_imag : out std_logic_vector(DIN_WIDTH-1 downto 0);
    out_sop : out std_logic;
    out_eop : out std_logic;
    out_valid : out std_logic
  );
end channel_estimator;
```

Ακροδέκτες Εισόδου

clk ▷ Το βασικό ρολόι του συστήματος.

rst ▷ Το βασικό σήμα επαναφοράς του συστήματος.

in_rst ▷ Είσοδος του σήματος ελέγχου *out_rst*. Σηματοδοτεί το τελευταίο data symbol του frame.

in_real ▷ Είσοδος δεδομένων εύρους 16 bit. Περιέχει το πραγματικό μέρος του μιγαδικού σήματος. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_imag ▷ Είσοδος δεδομένων εύρους 16 bit. Περιέχει το φανταστικό μέρος του μιγαδικού σήματος. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_sop ▷ Είσοδος που διεκπεραιώνει το ρόλο *startofpacket* του Avalon ST Interface.

in_eop ▷ Είσοδος που διεκπεραιώνει το ρόλο *endofpacket* του Avalon ST Interface.

in_valid ▷ Είσοδος που διεκπεραιώνει το ρόλο *sink valid* του Avalon ST Interface.

Ακροδέκτες Εξόδου

out_real ▷ Έξοδος δεδομένων εύρους 16 bit. Περιέχει το πραγματικό μέρος του μιγαδικού σήματος στο οποίο έχει γίνει πλέον -μετά από εκτίμηση καναλιού- ισοστάθμιση. Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_imag ▷ Έξοδος δεδομένων εύρους 16 bit. Περιέχει το φανταστικό μέρος του μιγαδικού σήματος στο οποίο έχει γίνει πλέον -μετά από εκτίμηση καναλιού- ισοστάθμιση. Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_valid ▷ Έξοδος που διεκπεραιώνει το ρόλο *source valid* του Avalon ST Interface.

out_sop ▷ Έξοδος που διεκπεραιώνει το ρόλο *startofpacket* του Avalon ST Interface.

out_eop ▷ Έξοδος που διεκπεραιώνει το ρόλο *endofpacket* του Avalon ST Interface.

Στον channel estimator επιτελείται η σύνθετη διαδικασία της ισοστάθμισης καναλιού. Το Α' στάδιο περιλαμβάνει την εκτίμηση καναλιού, ενώ το Β' την καθαυτό ισοστάθμιση. Η αρχή του κάθε ωφέλιμου OFDM frame σηματοδοτείται από το σήμα *in_sop* και το πέρας του από το *in_eop*. Το σήμα *in_rst* σηματοδοτεί το πέρας του frame και είναι αυτό που συγχρονίζει την εναλλαγή μεταξύ των σταδίων εκτίμησης και ισοστάθμισης.²⁷

Τώρα, στο Α' στάδιο, η εκτίμηση καναλιού πραγματοποιείται με τη χρήση ενός αλγορίθμου που περιλαμβάνει μια παραλλαγή της μεθόδου των *Ελαχίστων Τετραγώνων* (Least Squares - LS). Η μέθοδος στηρίζεται στην παρουσία του τρίτου, ειδικά διαμορφωμένου, training symbol, που περιέχει πιλοτικούς τόνους σε όλα τα subcarrier. Το σύμβολο αυτό είναι "γνωστό" σε αυτό το module. Μετά το πέρας αυτού του σταδίου έχει προκύψει η εκτίμηση καναλιού, η οποία τροφοδοτείται στο Β' στάδιο. Φυσικά το training symbol αφαιρείται από τη ροή των συμβόλων και δεν περνάει στο Β' στάδιο.

Στο Β'στάδιο λαμβάνει χώρα η ισοστάθμιση καναλιού στα data symbols του frame. Τα ισοσταθμισμένα στο κανάλι (μιγαδικά) subsymbol παρέχονται στις εξόδους *out_real* και *out_imag*, ενώ οι έξοδοι *out_sop* και *out_eop* σηματοδοτούν την αρχή και το πέρας, αντίστοιχα, του ωφέλιμου (useful) OFDM συμβόλου στη συχνότητα. Σε αυτό το σημείο, ιδανικά, το αρχικό σήμα πληροφορίας έχει απαλλαχτεί από τις ανεπιθύμητες αλλοιώσεις που προήλθαν κατά τη μετάδοσή του μέσα από μη ιδανικό κανάλι.

Από τη σκοπιά του συστήματος, τώρα, ο channel estimator εισάγει frontpressure στα modules του downstream, ακριβώς λόγω της ύπαρξης του σταδίου της εκτίμησης καναλιού. Μετά το πέρας του (και μέχρι το τέλος του frame) η ροή είναι συνεχής. Η καθυστέρηση E/E (I/O Latency) είναι μηδενική.

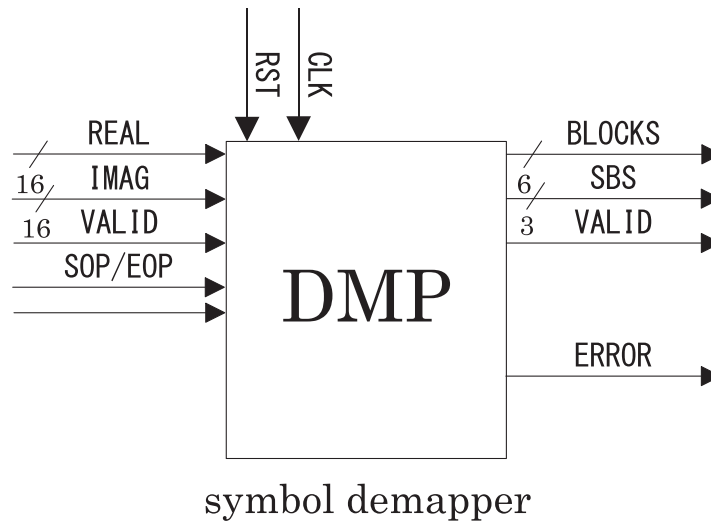
Παράμετρος	Σύμβολο	Περιγραφή
<i>FFT_SIZE</i>	<i>N</i>	Μέγεθος FFT και αριθμός subcarrier

Πίνακας 3.13: Παράμετροι του channel estimator

²⁷Μια μέθοδος παραγωγής του θα μπορούσε να είναι (εσωτερικά) από τα σήματα *in_sop*, *in_eop*, ωστόσο επιλέχθηκε να χρησιμοποιηθεί η υπάρχουσα δομή του *fine frequency estimator*.

Symbol Demapper

Στο Σχήμα 3.26 παρουσιάζεται το διάγραμμα ακροδεκτών του Symbol Demapper και ακολουθεί μια συνοπτική περιγραφή του κάθε ακροδέκτη.



Σχήμα 3.26: Διάγραμμα ακροδεκτών Symbol Demapper

```
entity symbol_demapper is
  generic(
    DIN_WIDTH_G : natural
  );
  port(
    clk, rst      : in std_logic;
    in_real       : in std_logic_vector(DIN_WIDTH_G-1 downto 0);
    in_imag       : in std_logic_vector(DIN_WIDTH_G-1 downto 0);
    in_valid      : in std_logic;
    in_sop, in_eop : in std_logic;
    out_blocks    : out std_logic_vector(5 downto 0);
    out_sbs       : out std_logic_vector(2 downto 0);
    out_valid     : out std_logic;
    out_error     : out std_logic
  );
end symbol_demapper;
```

Ακροδέκτες Εισόδου

clk ▷ Το βασικό ρολόι του συστήματος.

rst ▷ Το βασικό σήμα επαναφοράς του συστήματος.

in_real ▷ Είσοδος δεδομένων εύρους 16 bit. Περιέχει το πραγματικό μέρος του μιγαδικού σήματος. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_imag ▷ Είσοδος δεδομένων εύρους 16 bit. Περιέχει το φανταστικό μέρος του μιγαδικού σήματος. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_sop ▷ Είσοδος που διεκπεραιώνει το ρόλο *startofpacket* του Avalon ST Interface.

in_eop ▷ Είσοδος που διεκπεραιώνει το ρόλο *endofpacket* του Avalon ST Interface.

in_valid ▷ Είσοδος που διεκπεραιώνει το ρόλο *sink valid* του Avalon ST Interface.

Ακροδέκτες Εξόδου

out_blocks ▷ Έξοδος δεδομένων εύρους 6 bit, με χρήσιμη πληροφορία (ανά QAM4|16|64) 2, 4 ή 6 bit αντίστοιχα. Πρόκειται για την αποδιαμορφωμένη κατά QAM πληροφορία. Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_sbs ▷ Έξοδος εύρους 3 bit, περιέχει την πληροφορία του τύπου της αποδιαμόρφωσης κατά QAM που έλαβε χώρα. Αναφέρεται στα τρέχοντα περιεχόμενα της εξόδου *out_blocks*.

out_valid ▷ Έξοδος που διεκπεραιώνει το ρόλο *source valid* του Avalon ST Interface.

out_error ▷ Έξοδος που διεκπεραιώνει το ρόλο *error* του Avalon ST Interface.

Όταν το μιγαδικό σήμα φτάσει εδώ, αποτελείται αποκλειστικά από data σύμβολα, καθώς τα training σύμβολα έχουν ήδη χρησιμοποιηθεί και αφαιρεθεί. Η αρχή του κάθε ωφέλιμου (useful) OFDM συμβόλου (στη συχνότητα), σηματοδοτείται από το σήμα *in_sor* και το πέρας του από το *in_eop*. Πρόκειται για το τελευταίο module στο οποίο έχει κάποια έννοια ο όρος *ipacket*. Τα *isymbol* (στις θύρες *in_real* και *in_imag*), αποτελούνται από data subsymbols προς αποδιαμόρφωση.

Η λειτουργία του symbol demapper είναι η αποδιαμόρφωση κατά QAM των εισερχόμενων subsymbols και η παραγωγή k-blocks. Το module αυτό γνωρίζει το μοτίβο διαμόρφωσης QAM των subcarrier (*mod_pattern*), το οποίο -όπως έχει ήδη αναφερθεί- είναι κοινό και εξίσου γνωστό τόσο στον πομπό όσο και στο δέκτη. Μετά τη διαδικασία της αποδιαμόρφωσης, από την έξοδο *out_blocks* παρεχονται οι αποδιαμορφωμένοι κατά QAM συρμοί δεδομένων (k-blocks).²⁸ Το ωφέλιμο εύρος k του εκάστοτε k-block εξαρτάται από τον τύπο διαμόρφωσης και η κωδικοποίησή του βρίσκεται σε πλήρη αντιστοιχία με την έξοδο *out_data* του *disassembler*. Πιο συγκεκριμένα (Πίνακας 3.14), ανάλογα με το είδος της (απο)διαμόρφωσης, ισχύει για τη μορφή του block:²⁹

QAM 4	XXXXDD
QAM 16	XXDDDD
QAM 64	DDDDDD

Πίνακας 3.14: Μορφή του block πληροφορίας ανά τύπο QAM

Όπως και στους *disassembler* και *symbol mapper* στον πομπό, ο λόγος για τον οποίο συμβαίνει αυτό είναι ότι στο προγραμματισμό hardware δεν υπάρχει η έννοια της θύρας με εύρος που καθορίζεται δυναμικά κατά το runtime. Ο σχεδιαστής οφείλει να προδιαγράψει το εύρος της θύρας σύμφωνα με την πιο απαιτητική κατάσταση. Για παράδειγμα, αν ο τύπος της διαμόρφωσης QAM ήταν σταθερός (π.χ. QAM 4) η έξοδος *out_blocks* θα είχε εύρος 2 bit. Εδώ, όμως, η ύπαρξη μοτίβου διαμόρφωσης επιβάλλει τον καθορισμό του εύρους της θύρας στα 6 bit, λόγω του QAM 64 που είναι και η πιο απαιτητική -από άποψη εύρους του k-block- κατάσταση.

Τώρα, προκειμένου να γίνει σωστά η διάκριση ανάμεσα σε ένα bit πλεονασμού και σε ένα bit πληροφορίας στο επόμενο module, το κάθε block συνοδεύεται από τον τύπο της διαμόρφωσης του. Η πληροφορία αυτή παρέχεται στην έξοδο *out_sbs*. Κωδικοποιείται, δε, αποθηκεύοντας τον αριθμό k ως unsigned integer.³⁰

Από συστημικής απόψεως, η ροή των blocks (μετά από μία αρχική καθυστέρηση ?? κύκλων ρολογιού) είναι συνεχής και ο symbol demapper δεν ασκεί backpressure ούτε frontpressure.

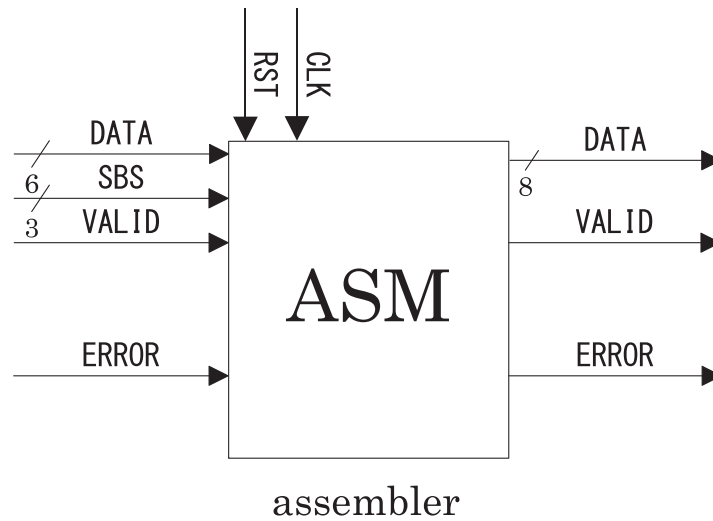
²⁸φυσικά το σήμα πληροφορίας παύει να είναι μιγαδικής φύσης μετά την αποδιαμόρφωση

²⁹το 'X' συμβολίζει το bit πλεονασμού (το οποίο δεν φέρει χρήσιμη πληροφορία) και το 'D' συμβολίζει το bit (ωφέλιμης) αποδιαμορφωμένης κατά QAM πληροφορίας.

³⁰Υπενθυμίζεται ότι η σχέση μεταξύ k και Q είναι $k = \log_2(Q)$

Assembler

Στο Σχήμα 3.27 παρουσιάζεται το διάγραμμα ακροδεκτών του Assembler και ακολουθεί μια συνοπτική περιγραφή του κάθε ακροδέκτη.



Σχήμα 3.27: Διάγραμμα ακροδεκτών Assembler

```
entity assembler is
  generic (
    DIN_WIDTH : natural;
    DOUT_WIDTH : natural
  );
  port (
    clk      : in std_logic;
    rst      : in std_logic;
    in_sbs   : in std_logic_vector(2 downto 0);
    -- Avalon Streaming Input (Sink) Interface
    in_data  : in std_logic_vector(DIN_WIDTH-1 downto 0);
    in_valid : in std_logic;
    in_error : in std_logic;
    -- Avalon Streaming Output (Source) Interface
    out_valid : out std_logic;
    out_data  : out std_logic_vector(DOUT_WIDTH-1 downto 0);
    out_error : out std_logic
  );
end assembler;
```

Ακροδέκτες Εισόδου

clk ▷ Το βασικό ρολόι του συστήματος.

rst ▷ Το βασικό σήμα επαναφοράς του συστήματος.

in_sbs ▷ Είσοδος εύρους 3 bit, περιέχει την πληροφορία του τύπου της αποδιαμόρφωσης κατά QAM που έλαβε χώρα στο προηγούμενο module. Αναφέρεται στα τρέχοντα περιεχόμενα της εισόδου *in_data*.

in_data ▷ Έξοδος δεδομένων εύρους 6 bit, με ωφέλιμο εύρος (ανά QAM4|16|64) 2, 4 ή 6 bit αντίστοιχα. Πρόκειται για την αποδιαμορφωμένη κατά QAM πληροφορία. Διεκπεραιώνει το ρόλο *sink data* του Avalon ST Interface.

in_valid ▷ Είσοδος που διεκπεραιώνει το ρόλο *sink valid* του Avalon ST Interface.

in_error ▷ Είσοδος που διεκπεραιώνει το ρόλο *error* του Avalon ST Interface.

Ακροδέκτες Εξόδου

out_valid ▷ Έξοδος που διεκπεραιώνει το ρόλο *source valid* του Avalon ST Interface.

out_data ▷ Έξοδος δεδομένων που έχουν ληφθεί, εύρους 8 bit (1 byte). Διεκπεραιώνει το ρόλο *source data* του Avalon ST Interface.

out_error ▷ Έξοδος που διεκπεραιώνει το ρόλο *error* του Avalon ST Interface.

Πρόκειται για το τελευταίο module της αλυσίδας του δέκτη. Η εργασία του συνίσταται στην ομαδοποίηση των k-blocks (των 2, 4 ή 6 bit) σε bytes. Από συστημική άποψη ασκεί frontpressure λόγω της διαδικασίας "συναρμολόγησης" του byte από μικρότερα δομικά στοιχεία (blocks). Από αυτό το module και πέρα αλλάζει το isymbol rate του συστήματος.

Κεφάλαιο 4

Μέσα στα μύρα κουτιά

4.1 Εισαγωγικά

Σε αυτό το κεφάλαιο θα γίνει εκτενής αναφορά στις λεπτομέρειες υλοποίησης όλων των module που απαρτίζουν το σύστημα, εφόσον η αρχιτεκτονική και η λειτουργική συμπεριφορά του συστήματος έχουν περιγραφεί αναλυτικά στο κεφάλαιο 3. Στο παρόν κεφάλαιο δε θα γίνει ιδιαίτερη περιγραφή της λειτουργίας του εκάστοτε module (τι), αλλά θα αναλυθεί η υλοποίησή του (πως).

Προς διευκόλυνση της επεξήγησης, σε κάθε module της αλυσίδας του datapath εκχωρείται ένα μοναδικό αναγνωριστικό αποτελούμενο από 3 λατινικά γράμματα. Τα ονόματα όλων των module που αποτελούν το σύστημα απεικονίζονται στο datapath του συστήματος (Σχήμα 3.4).

TX modules

4.2 Disassembler [DSA]

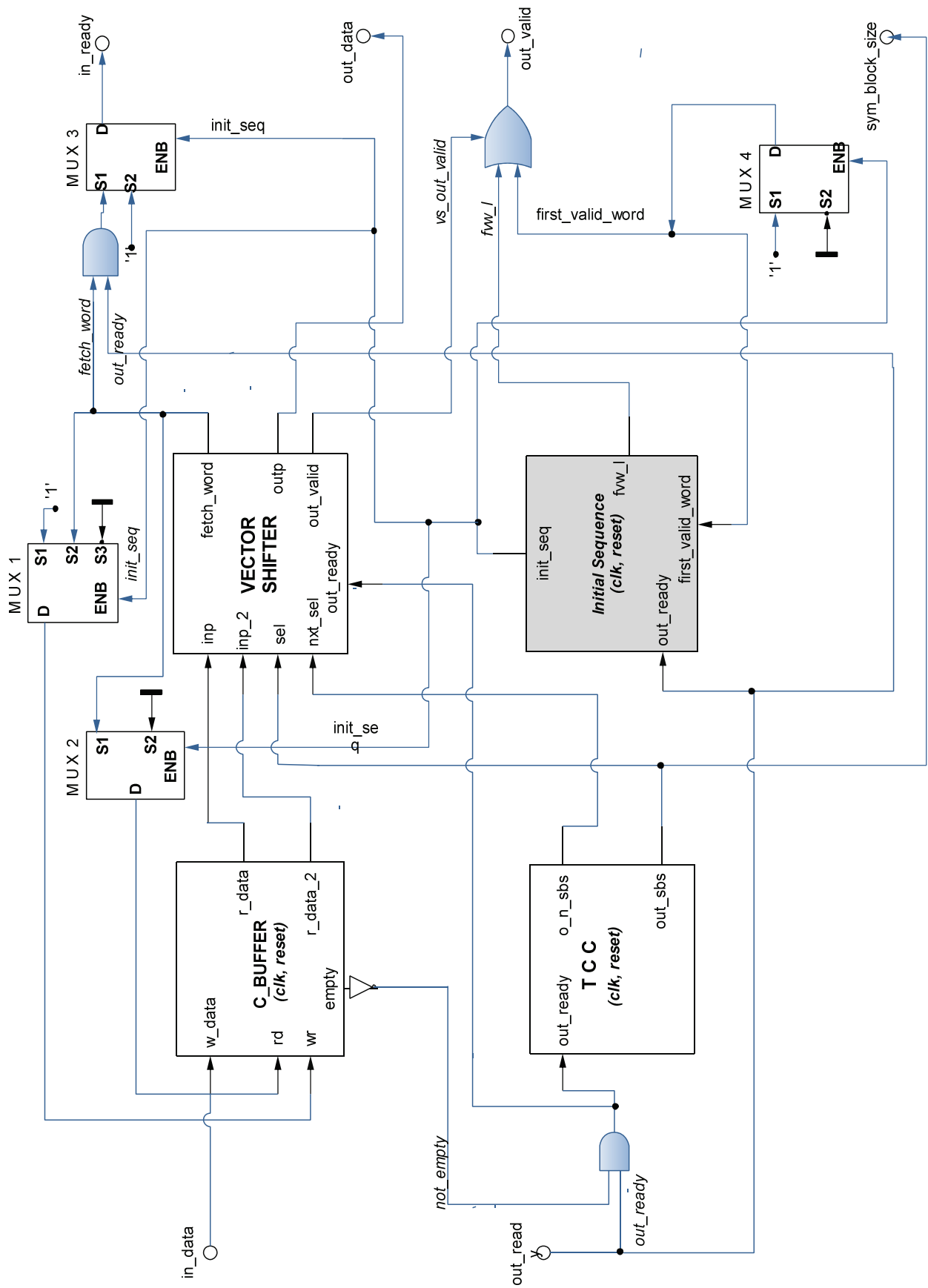
4.2.1 Η Top-Level Όψη

Στο σχήμα 4.1 παρουσιάζεται το top-level διάγραμμα του disassembler. Όπως αναφέρθηκε και στο κεφάλαιο 3, ο σκοπός του disassembler είναι η αποδόμηση των byte δεδομένων εισόδου σε συρμούς (σταθερού) ολικού μήκους 6 bit και ωφέλιμου μήκους που εξαρτάται από τον τύπο διαμόρφωσης κατά QAM, σύμφωνα τον κανόνα του Πίνακα 4.1, όπου το X συμβολίζει τα bit πλεονασμού και το D τα bit ωφέλιμης πληροφορίας.

QAM4	XXXXDD
QAM16	XXDDDD
QAM64	DDDDDD

Πίνακας 4.1: Κατανομή ωφέλιμων και πλεονάζοντων bit ανά τύπο διαμόρφωσης QAM

Η μεγαλύτερη πρόκληση είναι το γεγονός ότι (μετά από αρχική καθυστέρηση 2 κύκλων) η ροή των δεδομένων εξόδου θα πρέπει να είναι συνεχής. Η σχεδιαστική δυσκολία έγκειται στο ότι ο ρυθμός εισόδου των δεδομένων είναι $8 \frac{\text{bit}}{\text{datacycle}}$, ενώ ο ωφέλιμος ρυθμός εξόδου (*payload rate*) είναι μεταβλητός $[2, 4, 6] \frac{\text{bit}}{\text{datacycle}}$. Καθώς δεν είναι αποδεκτή η εισαγωγή φυσαλλίδων (bubbles) στη ροή των δεδομένων σε ένα Τ/Π σύστημα, κρίθηκε απαραίτητη η υλοποίηση ενός αντισταθμιστικού μηχανισμού για τις αντικρουόμενες αυτές απαιτήσεις.



Σχήμα 4.1: DSA Top-Level διάγραμμα

Τον πυρήνα του μηχανισμού αυτού (αλλά και του ίδιου του module) αποτελούν δύο δομές, ο *κυκλικός ενταμιευτής* (cyclic buffer) και ο *ολισθητής διανύσματος* (vector shifter), ο συνδυασμός των οποίων επιφέρει το επιθυμητό αποτέλεσμα, όπως θα φανεί παρακάτω.

4.2.2 Περιγραφή Κυκλωματικής Λειτουργίας

Η διεργασία *Initial Sequence*

Κατά τη διαδικασία εκκίνησης του συστήματος και ανά κύκλο ρολογιού, το σύστημα διέρχεται από τις εξής (κωδικοποιημένες) καταστάσεις

$$0(AK) \rightarrow 100_2, 1 \rightarrow 010_2, 2 \rightarrow 001_2, 3 \rightarrow 000_2, 4 \rightarrow 000_2, \dots$$

όπου ο αριθμός αριστερά του βέλους υποδηλώνει τον A/A του κύκλου ρολογιού. Οι δυαδικές αυτές τιμές αποτυπώνονται στο σήμα (εύρους 3 bit) *init_seq*, το οποίο αποτελεί το σήμα-επιλογή των τεσσάρων top-level πολυπλεκτών του module (MUX-1, MUX-2, MUX-3, MUX-4). Ο ρολος του καθενός από αυτούς τους πολυπλέκτες θα ξεκαθαριστεί στη συνέχεια. Προς το παρόν, αρκεί να αναφερθεί ότι σκοπός της διεργασίας αυτής, είναι η ομαλή και χωρίς παρενέργειες (όσον αφορά στην ακεραιότητα της διαχείρισης των δεδομένων) εκκίνηση του συστήματος. Ο ρόλος της τερματίζεται από τον 3ο κύκλο ρολογιού και πέρα, με την κατάσταση αυτή να αποτυπώνεται στην τιμή 000_2 του σήματος *init_seq*.

Παράλληλα, η διεργασία αυτή λειτουργεί και ως διαχειριστής της κατάστασης της latched εκδοχής του σήματος *first_valid_wire* το οποίο αποτελεί μέρος του μηχανισμού ελέγχου της εγκυρότητας των δεδομένων εξόδου του disassembler (μέσω του σήματος *out_valid*), όπως θα φανεί στη συνέχεια. Το σήμα αυτό παράγεται από τον ακόλουθο πολυπλέκτη (MUX-4), για τον οποίο γίνεται αναλυτικότερη αναφορά σε επόμενο εδάφιο:

```
with init_seq select
  first_valid_wire <= '1' when "001", '0' when others;
```

MUX-1

Πολυπλεκει το σήμα *fetch_word* (έξοδος του vector_shifter) και τις λογικές σταθερές 0 (GND) και 1 (V_{cc}) με βάση την τιμή του σήματος επιλογής *init_seq*. Η έξοδος του πολυπλέκτη αυτού είναι το σήμα *wr*, το οποίο ελέγχει τη διαδικασία εγγραφής δεδομένων (write) στον *c_buffer*.

```
-- cb_wr
with init_seq select
  cb_wr <= vs_fetch_word when "000",
  '0' when "100",
  '1' when others;
```

Η κυκλωματική συμπεριφορά του εν λόγω πολυπλέκτη συνίσταται στην απόδοση (στο σήμα *wr*) της τιμής 0 κατά την αρχική κατάσταση (AK, initial reset), της τιμής 1 για τους δύο πρώτους κύκλους ρολογιού και στο κλείδωμα στην εκάστοτε τιμή του σήματος *fetch_word* για το υπόλοιπο της λειτουργίας του συστήματος.

MUX-2

Μοιράζεται την ίδια λογική με τον MUX-1 αλλά, εδώ, για το σήμα *rd*, το οποίο ελέγχει τη διαδικασία ανάγνωσης (read) του *c_buffer*. Η πολυπλεξία που λαμβάνει χώρα αφορά στη λογική σταθερά 0 και στο σήμα *fetch_word*.

```
-- cb_rd
with init_seq select
  cb_rd <= vs_fetch_word when "000",
    '0' when others;
```

Κυκλωματικά, ο σκοπός του είναι να αποδίδει στο σήμα *rd* την εκάστοτε τιμή του *fetch_word*, από τον τρίτο κύκλο ρολογιού και πέρα. Πιο πριν η τιμή του *rd* είναι καθηλωμένη στο λογικό '0'.

Στον cyclic buffer

Πρόκειται για την κλασική δομή ενός circular buffer με τη διαφορά ότι, εδώ, εκτός από τη θύρα εισόδου *w_data* και τη θύρα εξόδου *r_data*, υπάρχει και μια δεύτερη θύρα εξόδου, η *r_data_2*. Η όλη λειτουργία εγγραφής και ανάγνωσης/διαγραφής δεδομένων συντονίζεται από τα σήματα ελέγχου *rd* και *wr* ενώ παρέχεται και ως έξοδος, μέσω του σήματος *empty*, η πληροφορία ότι ο buffer είναι άδειος και μπορεί να δεχτεί δεδομένα. Το μήκος του είναι 2 λέξεις εύρους 8 bit η καθεμία, άρα η συνολική του χωρητικότητα είναι 2 byte. Φυσικά, το εύρος της θύρας εισόδου δεδομένων είναι 8 bit, όσο ακριβώς είναι και το εύρος των δύο εξόδων.

Ο TX Constellation Controller

Ο ελεγκτής αυτός είναι υπεύθυνος για την ορθή αποστολή, στον vector shifter, του τύπου διαμόρφωσης QAM (μέσω του σήματος *out_sbs*) που έχει επιλεγθεί (μέσω του *mod_pattern*)¹ για το εκάστοτε k-block. Πλήν αυτού, παρέχει και τον τύπο διαμόρφωσης του επόμενου k-block, μέσω του σήματος *o_nxt_sbs*. Η διαδικασία αυτή επιτελείται συνεχώς εφόσον ισχύει η λογική συνθήκη

$$(empty)' \cdot out_ready = 1$$

ενώ, σε κάθε άλλη περίπτωση τίθεται σε αναμονή (stall). Με αυτό τον τρόπο υλοποιείται ο μηχανισμός αποδοχής backpressure σε αυτό το sub-module.

Ο Ολισθητής Διανύσματος - Vector Shifter

Σκοπός αυτού του sub-module είναι η παραγωγή των k-blocks με βάση την τιμή του *sbs*. Πρόκειται για ένα συνδυαστικό κύκλωμα το οποίο επιτελεί τη λειτουργία της ολίσθησης των δεδομένων εισόδου. Πέραν των θυρών ελέγχου, έχει δύο θύρες εισόδου δεδομένων, εύρους 1 byte (*inp* και *inp_2*) και μία θύρα εξόδου δεδομένων εύρους 6 bit (*outp*).

MUX-3

Ο πολυπλέκτης αυτός είναι υπεύθυνος για την υλοποίηση του μηχανισμού backpressure, που υπό προϋποθέσεις επιβάλλει ο disassembler στο upstream. Συγκεκριμένα, το σήμα επιλογέας είναι το *init_seq* και οι δύο εισοδοί είναι αφ'ενός η λογική σταθερά '1' (V_{cc}) και αφ'ετέρου το σήμα

$$out_ready \cdot fetch_word$$

```
-- dm_in_ready
with init_seq select
  dm_in_ready <= (out_ready and vs_fetch_word) when "000",
    '1' when others;
```

¹3, §3.1.3

Ο κυκλωματικός του ρόλος είναι να καθιστά το module σε κατάσταση ready, όταν τόσο η είσοδος *out_ready* (που εκφράζει το backpressure από το downstream) όσο και το σήμα *fetch_word* (που εκφράζει την «επιθυμία» του *Vector Shifter* για καινούρια δεδομένα) είναι ενεργοποιημένα. Σε αντίθετη περίπτωση το module επιβάλλει backpressure (*in_ready* = '0'). Αυτά ισχύουν για την κανονική λειτουργία (*init_seq* = '000'). Σε κάθε άλλη περίπτωση (αρχικά στάδια λειτουργίας) θα ισχύει (*in_ready* = '1') και το module μπορεί να δεχτεί δεδομένα.

MUX-4

Ο πολυπλέκτης αυτός αποτελεί βασικό τμήμα της υλοποίησης του μηχανισμού frontpressure που, υπό προϋποθέσεις, επιβάλλει ο disassembler στο downstream. το σήμα *out_valid*, συγκεκριμένα, το οποίο αποτελεί το «φορέα» του frontpressure, αποτιμάται σύμφωνα με τη λογική έκφραση

$$out_valid = (vs_out_valid) + (first_valid_wire) + (first_valid_wire_latch)$$

Το ακόλουθο snippet κώδικα δείχνει τη λειτουργία του πολυπλέκτη και τον τρόπο παραγωγής του τελικού σήματος *out_valid*.

```
with init_seq select
  first_valid_wire <= '1' when "001",
                    '0' when others;
.....
out_valid <= vs_out_valid or first_valid_wire or first_valid_wire_latch;
```

4.2.3 Τα Επιμέρους Υποσυστήματα

DSA.CYCLIC_BUFFER

```
component c_buffer is
  generic (
    B : natural :=8; -- number of bits
    W : natural :=4  -- number of address bits
  );
  port (
    clk, reset : in std_logic;
    rd, wr      : in std_logic;
    w_data      : in std_logic_vector ( B-1 downto 0);
    empty, full : out std_logic;
    r_data, r_data_2 : out std_logic_vector (B-1 downto 0)
  );
end component;
```

Στον buffer εισέρχονται τα byte των δεδομένων από την πηγή πληροφορίας (data source), ένα σε κάθε κύκλο δεδομένων (data cycle). Αυτή είναι η κατάσταση ομαλής λειτουργίας του *c_buffer* και επιτυγχάνεται με την απόδοση στα σήματα *rd* και *wr* της τιμής 1.

Η όλη ιδέα πίσω από την υιοθέτηση και δεύτερης εξόδου δεδομένων, είναι η υλοποίηση ενός «εσωτερικού» μηχανισμού *ping-pong buffer*.² Έτσι, σε κάθε κύκλο ρολογιού, εναλλάσσονται οι θύρες *r_data* και *r_data_2* στο ρόλο της κύριας θύρας εξόδου των δεδομένων. Ο σκοπός του μηχανισμού αυτού θα γίνει κατανοητός κατά την περιγραφή του *vector_shifter*.

Για την πληρότητα της περιγραφής, στον πίνακα 4.2 παρατίθενται οι δυνατοί συνδυασμοί τιμών των σημάτων ελέγχου *rd* και *wr* και μια συνοπτική ερμηνεία της λειτουργίας του buffer ανά περίπτωση.

²Ο ping-pong buffer είναι ένας buffer μετάδοσης ο οποίος αποτελείται από δύο ξεχωριστούς buffer με εναλλασσόμενη λειτουργία. Ενώσω ο ένας buffer δέχεται την καινούρια μεταδιδόμενη πληροφορία, ο άλλος διαβάζει την παλιά.

Σήμα wr	Σήμα rd	Λειτουργία buffer
0	0	Ο buffer είναι αδρανής. Αυτό μπορεί να συμβεί στην ΑΚ ή όταν, στην κανονική λειτουργία, ο vector_shifter δε χρειάζεται καινούριο byte δεδομένων.
0	1	Αδύνατη συγκυρία.
1	0	Λαμβάνει χώρα μόνο εγγραφή δεδομένων και όχι ανάγνωση/διαγραφή. Απαντάται στους 2 πρώτους κύκλους ρολογιού από την απαρχή της λειτουργίας του συστήματος.
1	1	Ο buffer σε πλήρη δράση. Γίνεται εγγραφή στη μια θέση, ενώ στην άλλη τα δεδομένα απελευθερώνονται μέσω της διαδικασίας ανάγνωσης. Αποτελεί τον κανονικό τρόπο λειτουργίας του buffer στην περίπτωση όπου ο vector_shifter έχει ενεργοποιημένη την απαίτηση εγγραφής καινούριου byte στον buffer (<i>fetch_word = '1'</i>)

Πίνακας 4.2: Τα σήματα rd και wr ως ρυθμιστές της λειτουργίας του c_buffer

DSA.VECTOR_SHIFTER

```

component shifter is
  port (
    clk, reset : in std_logic;
    inp, inp_2 : in std_logic_vector(7 downto 0);
    sel, nxt_sel : in integer range 0 to 7;
    out_ready : in std_logic;
    out_valid : out std_logic;
    outp : out std_logic_vector(5 downto 0);
    fetch_word : out std_logic
  );
end component;

```

Επιλέχτηκε η υλοποίησή του ως συνδυαστικό κύκλωμα και όχι ως ακολουθιακό (όπου θα υπήρχε εξοικονόμηση υλικού), λόγω των αυστηρών χρονικών περιορισμών που επιβάλλονται στον disassembler. Στο υλικό, αυτή η δομή αποτυπώνεται ως δύο σύνολα των 9 ψευδο-καταχωρητών (pseudo-registers) το καθένα, με τον καθένα να έχει μήκος 8 bit. Κάθε «καταχωρητής» σε ένα σύνολο, ανάλογα με τη σχετική του απόσταση από τον «καταχωρητή» στον οποίο πρωτοεμφανίζεται το byte εισόδου (αρχικός), περιέχει μια ολισθημένη εκδοχή αυτού του byte. Το βήμα ολίσθησης είναι 1, δηλαδή κάθε «καταχωρητής» περιέχει μία, κατά ± 1 bit, ολισθημένη εκδοχή του αρχικού διανύσματος, σε σχέση με τους εκατέρωθέν του. Τώρα, η κάθε είσοδος δεδομένων του vector shifter αντιστοιχίζεται στους αρχικούς «καταχωρητές» των δύο συνόλων.

Εδώ είναι σημαντικό να τονιστεί ότι οι «καταχωρητές» του πρώτου συνόλου (πλην του αρχικού), περιέχουν όχι μόνο την εκάστοτε ολισθημένη εκδοχή του αρχικού διανύσματος (byte), αλλά και έναν αριθμό από bit της δεύτερης λέξης (στο τέλος τους), τόσα όσος και ο βαθμός ολίσθησής τους. Αντίστοιχα, οι «καταχωρητές» του δεύτερου συνόλου (πλην του αρχικού), περιέχουν την ολισθημένη εκδοχή του δεύτερου byte και έναν αριθμό αδιάφορων bit (λογικό 'X') στο τέλος τους, τόσα όσος και ο βαθμός ολίσθησής τους. Το παρακάτω snippet κώδικα απεικονίζει την υλοποίηση των παραπάνω:

```

row(0) <= inp;
row_2(0) <= inp_2;
G1: for i in 1 to 8 generate --7
  row(i) <= row(i-1) (6 downto 0) & row_2(i-1)(7);
end generate;
G2: for i in 1 to 8 generate --7
  row_2(i) <= row_2(i-1) (6 downto 0) & 'X';
end generate;

```


Η έξοδος του κυκλώματος επιλέγεται κάθε φορά, ως ένα τμήμα κάποιου «καταχωρητή» του πρώτου συνόλου (τρέχων «καταχωρητής»). Η βασική ιδέα πίσω από τη λειτουργία του vector shifter μπορεί να γίνει ευκολότερα κατανοητή με τη μελέτη της λειτουργίας του ανά τύπο διαμόρφωσης κατά QAM.

QAM-4 (QPSK)

Σε αυτή την περίπτωση τα ωφέλιμα bit πληροφορίας είναι 2. Έτσι λαμβάνονται ως ωφέλιμη έξοδος τα 2 πρώτα bit του byte που βρίσκεται στον αρχικό "καταχωρητή" και "ενεργοποιείται" (δηλαδή γίνεται τρέχων) ο "καταχωρητής" που περιέχει την ολισθημένη κατά 2 θέσεις εκδοχή του αρχικού byte.

QAM-16

Σε αυτή την περίπτωση τα ωφέλιμα bit πληροφορίας είναι 4. Όπως και πάνω, λαμβάνονται ως ωφέλιμη έξοδος τα 4 πρώτα bit και "ενεργοποιείται" ο "καταχωρητής" που περιέχει το ολισθημένο κατά 4 θέσεις byte.

QAM-64

Σε αυτή την περίπτωση τα ωφέλιμα bit πληροφορίας είναι 6. Με την ίδια ακριβώς λογική, την ωφέλιμη έξοδο αποτελούν τα πρώτα 6 bit και "ενεργοποιείται" ο αντίστοιχος "καταχωρητής" 6 θέσεις πιο μακριά.

Τώρα, αν το άθροισμα των διαδοχικών τύπων QAM ήταν πάντα 8, π.χ. {2,2,2,2} ή {2,4,2} ή {4,4} ή {6,2} κλπ., τότε τα πράγματα θα ήταν αρκετά πιο εύκολα δεδομένου ότι τότε δεν θα απαιτείτο το δεύτερο σύνολο "καταχωρητών". Πράγματι, θα αρκούσε το πρώτο σύνολο καθώς σε αυτό θα αποτυπώνονταν όλες οι δυνατές μορφές εξόδου και με έναν απλό έλεγχο της τιμής του σήματος *sel* θα καθοριζέτο αμέσως ο κατάλληλος, κάθε φορά, "καταχωρητής" από τον οποίο θα γινόταν η λήψη της ωφέλιμης πληροφορίας. Όμως, το *mod_pattern* είναι πολύ πιο ευέλικτη δομή και επιτρέπει όλους τους δυνατούς συνδυασμούς διαμορφώσεων κατά QAM. Έτσι προκύπτουν υπο-ακολουθίες διαμορφώσεων όπως οι ακόλουθες: {6,4} {6,6} {4,6,6} κλπ. με αθροίσματα αξιών που ξεπερνούν τη "βολική" τιμή των 8 bit(1 byte). Αυτός είναι ο λόγος για τον οποίο απαιτείται η δεύτερη είσοδος στον vector shifter και, εν τέλει, και το δεύτερο σύνολο "καταχωρητών". Εδώ, όμως, έρχεται προς βοήθεια το σήμα *nxt_sel* το οποίο περιέχει την πληροφορία του τύπου διαμόρφωσης του επόμενου (κάθε φορά) k-block (γνωστός λόγω του καθορισμένου -κάθε φορά- *mod_pattern*). Έτσι είναι εφικτός ένας ακριβής προϋπολογισμός των απαιτήσεων σε δεδομένα. Η απαίτηση του vector shifter για καινούρια δεδομένα, "διατυπώνεται" στον buffer μέσω του σήματος *fetch_word* έγκαιρα, έτσι ώστε να υπάρχει χρόνος να φορτωθούν τα καινούρια δεδομένα χωρίς να μπει το σύστημα σε αναμονή με δυσάρεστο επακόλουθο τη δημιουργία φουσαλλίδων στη ροή των δεδομένων.

Τέλος, πρέπει να αναφερθεί ότι όλη η παραπάνω διαδικασία τίθεται σε αναμονή(stall) όταν και μόνο όταν ισχύει η συνθήκη

$$(empty)' \cdot out_ready = 0$$

Η τιμή αυτής της συνθήκης διαβιβάζεται στην είσοδο *out_ready* του vector shifter και η λογική της ερμηνεία είναι ότι η διαδικασία σταματά προσωρινά όταν το σήμα *out_ready* του top-level του disassembler είναι 0 (εφαρμογή δηλαδή backpressure από το downstream) ή ο buffer είναι πλήρως αδειος (κάτι που συμβαίνει μόνο κατά την αρχικοποίηση της μετάδοσης) ή και τα δύο μαζί.

DSA.TX_CONSTELLATION_CONTROLLER

```
component tx_const_control is
  generic (
    DELAY : natural := 0 -- TOTAL EXTERNAL FEEDBACK-CHAIN DELAY
  );
  port (
    clk : in std_logic;
    rst : in std_logic;
```

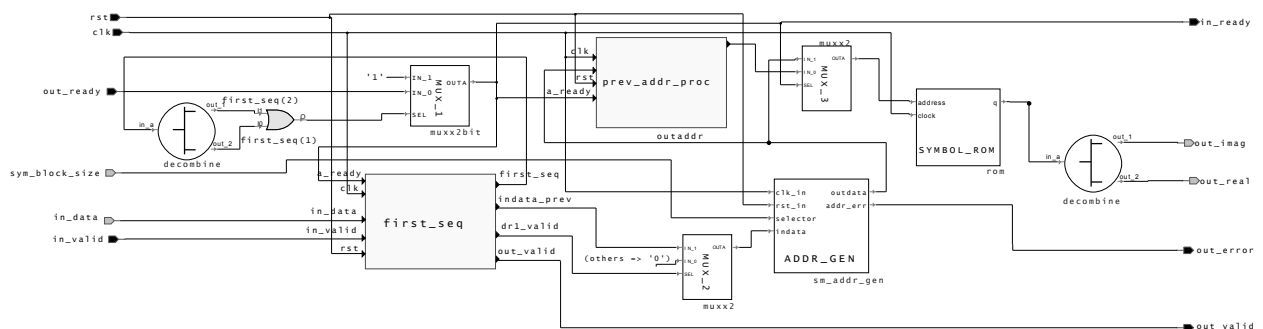
```

-- Output backpressure
out_ready : in std_logic;
out_sbs : out std_logic_vector (2 downto 0); -- generated symblocksize (2, 4, 6)
o_nxt_sbs : out std_logic_vector (2 downto 0) -- next sbs
);
end component;

```

4.3 Symbol Mapper [SMA]

4.3.1 Η Top-Level Όψη



Σχήμα 4.2: MAP Top-Level διάγραμμα

Στο σχήμα 4.2 παρουσιάζεται το top-level διάγραμμα του symbol mapper.

Πρόκειται για ένα σχετικά απλό module, η κύρια λειτουργία του οποίου συνίσταται στη μετατροπή των k-blocks των δεδομένων εισόδου σε, διαμορφωμένα κατά QAM, OFDM subsymbols. Εδώ, δηλαδή είναι που λαμβάνει χώρα η διαμόρφωση κατά QAM των δεδομένων εισόδου.

Από κυκλωματικής απόψεως, τα βασικά συστατικά του στοιχείου είναι μία μνήμη ROM (*SYMBOL_ROM*) στην οποία βρίσκονται αποθηκευμένα τα σύμβολα (σε μιγαδική αναπαράσταση) των αστερισμών QAM και ένα υποσύστημα παραγωγής της διεύθυνσης των κατάλληλων, κάθε φορά, συμβόλων QAM στη ROM (*ADDR_GEN*).

4.3.2 Περιγραφή Κυκλωματικής Λειτουργίας

Τα k-blocks φτάνουν στο module μέσω του ακροδέκτη εισόδου *in_data* και μαζί με τα σήματα ελέγχου *a_ready* (έξοδος του MUX_1) και *in_valid* "εισέρχονται" στο τμήμα-διεργασία *first_seq*. Σκοπός της διεργασίας αυτής είναι να εξασφαλίσει την ορθή λειτουργία του κυκλώματος κατά τους πρώτους κύκλους μετά την έναρξη λειτουργίας του κυκλώματος.

Η διεργασία *first_seq*

Υπάρχουν τέσσερις καταστάσεις από τις οποίες "περνα" διαδοχικά το κύκλωμα κατά τη διαδικασία εκκίνησης (Πίνακας 4.3). Κωδικοποιούνται μέσω του σήματος ελέγχου *first_seq* το οποίο αποτελεί μία από τις άτυπες εξόδους της διεργασίας.

Σε αυτό το τμήμα υλοποιείται, με χρήση μανδαλωτών και κατάλληλες αποδόσεις τιμών, τόσο ο μηχανισμός frontpressure από τα modules του upstream, όσο και ο μηχανισμός backpressure από τα modules του downstream.

Τιμή <i>first_seq</i>	Περιγραφή
100 ₂	Αρχική Κατάσταση (ΑΚ) • Εμφανίζεται σε επόμενες καταστάσεις όταν δεν υπάρχουν έγκυρα δεδομένα εισόδου • <i>in_ready</i> = 1
010 ₂	Δεύτερος κύκλος ρολογιού από την έναρξη λειτουργίας • <i>in_ready</i> = 1
001 ₂	Τρίτος κύκλος ρολογιού από την έναρξη λειτουργίας • <i>in_ready</i> = <i>out_ready</i>
000 ₂	Συνηθισμένη Κατάσταση Λειτουργίας • Έγκυρα δεδομένα εισόδου • <i>in_ready</i> = <i>out_ready</i>

Πίνακας 4.3: Εκκίνηση και λειτουργία κυκλώματος

Τις υπόλοιπες άτυπες εξόδους αποτελούν τα εξής σήματα :

indata_prev ▷ Είναι η "μανδαλωμένη" (latched) εκδοχή των δεδομένων εισόδου *in_data* με την τιμή του να εξαρτάται από την τιμή των σημάτων *in_valid* και *a_ready*

dr1_valid ▷ Αποτελεί την κατά 1 κύκλο ρολογιού καθυστερημένη εκδοχή του σήματος εισόδου *in_valid*

dr2_valid ▷ Αποτελεί την κατά 2 κύκλους ρολογιού καθυστερημένη εκδοχή του σήματος εισόδου *in_valid*. Το σήμα *dr2_valid* προκύπτει από την ανάγκη ευθυγράμμισης (alignment) του σήματος *out_valid* με τα δεδομένα εξόδου (*out_real*, *out_imag*).

MUX_2

Πολυπλέκει το σήμα *indata_prev* και τη λογική σταθερά '0' (γείωση κυκλώματος) με βάση το σήμα ελέγχου *dr1_valid*. Η έξοδος (*temp_data*) τροφοδοτείται στην είσοδο δεδομένων *indata* του υποσυστήματος παραγωγής διευθύνσεων (address generator).

```
--MUXed signal (ROM Input Sanity Check)
temp_data <= indata_prev when ((dr1_valid = '1') ) else (others => '0');
```

Χρησιμοποιείται για την διασφάλιση της εγκυρότητας των δεδομένων εισόδου στον address generator.

Στον address generator

Η άλλη είσοδος του *ADDR_GEN* είναι το σήμα *sym_block_size* (εισέρχεται διαμέσου του ακροδέκτη *selector*) και περιέχει την πληροφορία του είδους διαμόρφωσης για το τρέχον k-block. Το αποτέλεσμα είναι η παραγωγή της κατάλληλης διεύθυνσης εισόδου (σύμφωνα με μια διαδικασία που θα περιγραφεί στο κατάλληλο εδάφιο) για τη ROM των συμβόλων (symbol ROM). Αυτή η διεύθυνση παρέχεται από το σήμα *outaddr* διαμέσου του ακροδέκτη εξόδου *out_data*.

Η διεργασία *prev_addr_proc*

Έχοντας ως άτυπη είσοδο ελέγχου το σήμα *a_ready* (έξοδος του *MUX_1*) η διεργασία αυτή υλοποιεί τμήμα του μηχανισμού backpressure που δέχεται το παρόν υποσύστημα από τα module του down-stream. Αποτελεί ένα latch για το σήμα *outaddr* (στην άτυπη έξοδο *outaddr_prev*).

MUX_3

Πολυπλέκει τη διεύθυνση εισόδου της ROM (στη σύγχρονη μορφή της - *outaddr* - ή στη μανδαλωμένη - *outaddr_prev* -) και τροφοδοτεί την αντίστοιχη είσοδο της ROM. Επιλογέας είναι το σήμα *a_ready*.

```
-- MUXed
outaddr_curr <= outaddr_prev when ((a_ready = '0')) else outaddr;
```

Εξασφαλίζει την εγκυρότητα της διεύθυνσης εισόδου στη ROM.

Στη ROM

Βάσει της κατάλληλα δομημένης διεύθυνσης εισόδου, παρέχεται στην έξοδο *q*, εύρους 32 bit, το μιγαδικό OFDM subsymbol. Το σήμα αυτό, διαμέσου ενός άτυπου "διαχωριστή σήματος" (signal splitter), διοχετεύεται στις εξόδους *out_real* και *out_imag*.

MUX_1

Πολυπλέκει το σήμα ελέγχου *a_ready* και τη λογική σταθερά '1', με επιλογή το σήμα *first_seq*. Έξοδος είναι το σήμα *a_ready*.

```
a_ready <= '1' when (first_seq(1) = '1' or first_seq(2) = '1') else out_ready;
```

Εξασφαλίζει την ορθή "αντιμετώπιση" και χρήση του σήματος ελέγχου *out_ready*, κατά το στάδιο της αρχικοποίησης του module.

4.3.3 Τα Επιμέρους Υποσυστήματα

MAP.ADDR_GEN

```
component sm_addr_gen is
  port (
    clk_in : in std_logic;
    rst_in: in std_logic;
    selector: in std_logic_vector (2 downto 0);
    indata : in std_logic_vector (5 downto 0); --indata : in std_logic_vector (INDATABUS-1 downto 0);
    outdata : out std_logic_vector (6 downto 0); --outdata : out std_logic_vector (OUTDATABUS downto 0);
    addr_err : out std_logic
  );
end component;
```

Τα μιγαδικά σύμβολα των αστερισμών QAM βρίσκονται τοποθετημένα στη ROM. Είναι ομαδοποιημένα ως εξής : Συνεχόμενα ανά τύπο QAM ενώ, μεταξύ των αρχικών διευθύνσεων των διαφορετικών τύπων, υπάρχει κατάλληλο offset. Πιο συγκεκριμένα, η διεύθυνση των συμβόλων στη ROM έχει τη μορφή που αναγράφεται στον Πίνακα 4.4 (ανά τύπο QAM) που καθορίζεται από το σήμα *sym_block_size* (εφεξής SBS).

Υπενθυμίζεται ότι τα δεδομένα εισόδου φέρουν, ανάλογα με τον τύπο QAM, 2 ή 4 bit πλεονασμού, τα οποία φυσικά δε λαμβάνονται υπόψη κατά τη διαδικασία σχηματισμού της διεύθυνσης.

Τέλος, το σήμα *addr_error* ενεργοποιείται σε περίπτωση που το σήμα *sbs* έχει λάβει τιμή διαφορετική από τις επιτρεπόμενες 010₂, 100₂, 110₂.

Τύπος QAM	Τιμή SBS	Offset	Τελική μορφή διεύθυνσης
QAM4	010 ₂	0000000 ₂	"0000000"+DD
QAM16	100 ₂	0000100 ₂	"0000100"+DDDD
QAM64	110 ₂	0010100 ₂	"0010100"+DDDDDD

Πίνακας 4.4: Σχηματισμός διεύθυνσης ROM ανά τύπο διαμόρφωσης QAM

MAP.SYMBOL_ROM

```

component rom is
  port (
    address    : IN STD_LOGIC_VECTOR (6 DOWNTO 0);
    clock      : IN STD_LOGIC := '1';
    q          : OUT STD_LOGIC_VECTOR (31 DOWNTO 0)
  );
end component;

```

Πρόκειται για μια "τυποποιημένη" Altera ROM, τα τεχνικά χαρακτηριστικά της οποίας αναγράφονται στον Πίνακα 4.5.

Τύπος	Altsyncrom Single Port
Εύρος θύρας διεύθυνσης	7 bit
Εύρος θύρας εξόδου δεδομένων	32 bit

Πίνακας 4.5: Τεχνικά Χαρακτηριστικά Altera ROM (MAP)

Μετά από καθυστέρηση 1 κύκλου ρολογιού και με βάση τη διεύθυνση που παρέχεται στην είσοδο *address*, εμφανίζονται στην έξοδο *q*, συνεπτυγμένα, το πραγματικό και το φανταστικό μέρος του μιγαδικού *sub-symbol*.

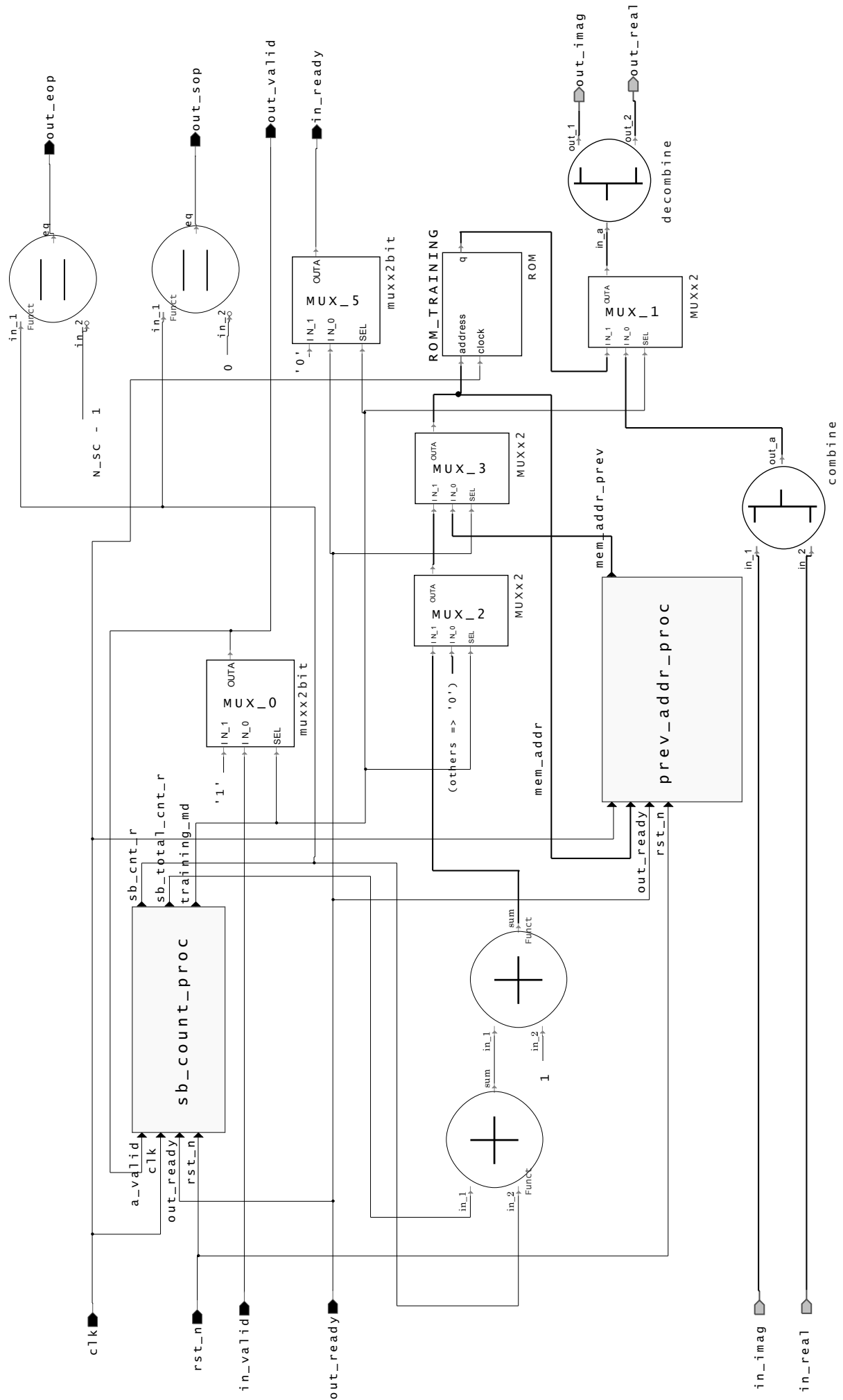
4.4 Frame Generator [FRG]

4.4.1 Η Top-Level Όψη

Στο σχήμα 4.3 παρουσιάζεται το top-level διάγραμμα του Frame Generator (FRG). Η κύρια λειτουργία του συνίσταται στην ομαδοποίηση των *subsymbol* σε OFDM σύμβολα και *frame*.

4.4.2 Περιγραφή Κυκλωματικής Λειτουργίας

Καθώς κάθε *frame* ξεκινάει με το προοίμιο, υπάρχει μία ROM στην οποία είναι αποθηκευμένα τα *training symbols* που το αποτελούν. Ένα ακολουθιακό κύκλωμα δύο συνδεδεμένων μετρητών (*process sb_count - proc*) καταγράφει τον αριθμό των *subsymbol* που έχουν περάσει και προσδιορίζει τα όρια του συμβόλου. Επιπλέον, καθορίζει το πότε θα εισαχθούν τα *training symbol* στην ακολουθία των συμβόλων, οριοθετώντας το *frame*. Κατά τη διάρκεια των 3 πρώτων συμβόλων το κύκλωμα βρίσκεται σε κατάσταση "training" και το *datapath* της εξόδου συνδέεται στην έξοδο της ROM μέσω ενός MUX. Το *in_ready* κατεβαίνει, ασκώντας *backpressure* στα *upstream modules*. Από το 4^ο μέχρι το *R* σύμβολο, το κύκλωμα βρίσκεται σε κατάσταση "data" και το *datapath* της εξόδου συνδέεται στην είσοδο, με το *in_ready* να συνδέεται πλέον στο *out_ready*.



Σύντομο 4.3: FPGA Top-level διάγραμμα

Η διεργασία sb_count_proc

Το κύκλωμα των μετρητών (process sb_count_proc) αποτελείται από δύο συνδεδεμένους σε αλυσίδα μετρητές έτσι ώστε, όταν "μηδενίζει" ο πρώτος, να αυξάνεται ο δεύτερος. Ο πρώτος (sb_cnt) μετράει subsymbol με όρια $(0 \dots N_o - 1)$, ενώ ο δεύτερος (sy_cnt) μετράει OFDM σύμβολα, με όρια $(0 \dots R - 1)$. Από τους παραπάνω μετρητές μόνο ο sb_cnt περνάει στην έξοδο του process στο σήμα sb_cnt_r, ενώ ο άλλος χρησιμοποιείται εσωτερικά. Συνδυαστική λογική παράγει τα σήματα ελέγχου του Avalon (out_sop, out_eop, out_valid, in_ready). Όταν ο sb_cnt_r είναι μηδέν σηκώνεται το out_sop, ενώ όταν είναι $N_o - 1$ σηκώνεται το out_eop, σηματοδοτώντας την αρχή και το τέλος του κάθε συμβόλου, αντίστοιχα. Εκτός αυτού το process παράγει και τον sb_total_cnr_r, ο οποίος μετράει συνεχόμενα subsysmol, μέχρι το τέλος του frame. Χρησιμοποιείται για την κατασκευή της διεύθυνσης μνήμης της ROM.

Η διεργασία prev_addr_proc

Εδώ λαμβάνεται ειδική μέριμνα για την υλοποίηση του backpressure με Ready Latency = 0. Καθώς η ROM έχει ένα κύκλο I/O latency, υπάρχει ένα βοηθητικό ακολουθιακό κύκλωμα (prev_addr_proc) για την συγκράτηση της προηγούμενης διεύθυνσης μνήμης. Η τελευταία, τροφοδοτείται στη μνήμη, όταν ασκείται backpressure από το downstream.

Ο MUX MUX_0

Στην έξοδό του παράγεται το σήμα out_valid, δηλαδή το fronpressure της εξόδου. Ανάλογα με την τιμή του training_md επιλέγεται :

- Η τιμή '1', όταν το κύκλωμα βρίσκεται σε training mode.
- Η τιμή του in_valid, όταν το κύκλωμα βρίσκεται σε data mode.

Ο MUX MUX_1

Συνδέει στο datapath της εξόδου είτε την έξοδο της ROM, είτε την είσοδο δεδομένων του κυκλώματος, ανάλογα με την τιμή του training_md

Ο MUX MUX_2

Οδηγεί την διεύθυνση της μνήμης ROM όταν το training_md = '1'. Σε αντίθετη περίπτωση τοποθετεί μηδενικά στο διάδρομο της μνήμης, ώστε η διεύθυνση της μνήμης να είναι σε κάθε περίπτωση, έγκυρη.

Ο MUX MUX_3

Αποτελεί τμήμα του μηχανισμού υλοποίησης του backpressure. Όταν το downstream δεν μπορεί να δεχτεί δεδομένα, (out_ready = '0'), οδηγείται στον διάδρομο διεύθυνσης της ROM η προηγούμενη τιμή του, όπως αυτή έχει μανδαλωθεί από το prev_addr_proc.

Ο MUX MUX_5

Οδηγεί το σήμα in_ready, δηλαδή το backpressure που ασκεί αυτό το module στο upstream. Συγκεκριμένα η τιμή αυτού του σήματος είναι :

- '0' , όταν το κύκλωμα βρίσκεται σε training mode
- συνδεδεμένη στο out_ready, όταν το κύκλωμα βρίσκεται σε data mode

4.4.3 Τα Επιμέρους Υποσυστήματα

FRG.ROM_TRAINING

```
component rom_training is
  port
  (
    address      : IN STD_LOGIC_VECTOR (11 DOWNTO 0);
    clock        : IN STD_LOGIC := '1';
    q            : OUT STD_LOGIC_VECTOR (31 DOWNTO 0)
  );
end component rom_training;
```

Πρόκειται για μια "τυποποιημένη" Altera ROM, τα τεχνικά χαρακτηριστικά της οποίας αναγράφονται στον Πίνακα 4.6.

Τύπος	altsyncram • Single Port
Εύρος θύρας διεύθυνσης	12 bit
Εύρος θύρας εξόδου δεδομένων	32 bit

Πίνακας 4.6: Τεχνικά Χαρακτηριστικά Altera ROM (FRG)

Μετά από καθυστέρηση 1 κύκλου ρολογιού και με βάση τη διεύθυνση που παρέχεται στην είσοδο address, εμφανίζονται στην έξοδο q, συνεπτυγμένα, το πραγματικό και το φανταστικό μέρος του training subsymbol.

Τα training symbol ανήκουν στα asset του κυκλώματος και η μορφή τους έχει να κάνει με τον συγχρονισμό (2.2.3) και την εκτίμηση καναλιού (2.2.5). Ο μηχανισμός Asset Dispatching είναι υπεύθυνος για την παραγωγή του αρχείου προγραμματισμού της ROM. Περισσότερα για τα asset αναφέρονται στην §3.1.3.

RX modules

4.5 Αριθμητική

Σε αυτό το σημείο θα γίνει μία σύντομη αναφορά στα θέματα αριθμητικής που αφορούν το κύκλωμα, καθώς είναι αναγκαία για την περιγραφή των ακόλουθων module.

Συστήματα αριθμητικής Είναι γνωστό, πως υπάρχουν διαφορετικά συστήματα αριθμητικής, που διακρίνονται σύμφωνα με τη βάση (b) και το αλφάβητο των ψηφίων τους.³ Δύο σημαντικές διευκρινήσεις:

³Τα συνηθισμένα συστήματα αριθμητικής ακεραίων είναι σταθερής βάσης, b , για όλα τα ψηφία τους και έχουν αλφάβητο $\mathcal{A} = [0, 1, 2, \dots, b-2, b-1]$. Ένας ακέραιος θετικός αριθμός (unsigned integer) αναπαρίσταται από μία συμβολοσειρά ψηφίων, $X \equiv X_{N-1}X_{N-2} \dots X_0$, $X_n \in \mathcal{A}$. Η αξία του αριθμού ορίζεται ως το άθροισμα της αξίας του κάθε ψηφίου επί την αξία της θέσης του ψηφίου. Η αξία της θέσης ενός ψηφίου είναι, από δεξιά προς τα αριστερά, $0, b, b^2, b^3, \dots$. Δηλαδή η αξία ενός N -ψηφίου αριθμού είναι:

$$V(X) = \sum_{k=0}^{N-1} X_k b^k$$

- α) ένας ακέραιος αριθμός μπορεί να αναπαρασταθεί ισοδύναμα σε καθένα από αυτά,
- β) ένας δεκαδικός αριθμός είναι πιθανό να μην αναπαρίσταται επ' ακριβώς σε ένα σύστημα αριθμητικής.

Εξέχουσα σημασία έχει το δυαδικό σύστημα, όπου $b = 2$, καθώς με αυτό γίνονται όλες οι πράξεις στα ψηφιακά συστήματα, όπως το παρόν. Το αλφάβητό του είναι $[0, 1]$ και τα ψηφία καλούνται bit (binary digits).

Δυαδική κωδικοποίηση Για την αποθήκευση ενός πραγματικού αριθμού, σε έναν δυαδικό ψηφιακό υπολογιστή, η αναπαράστασή του πρέπει να κωδικοποιηθεί σε κατάλληλη μορφή. Αυτή, θα πρέπει να είναι αποδοτική ως προς το χώρο, να καλύπτει τις απαιτήσεις σε ακρίβεια και δυναμικό εύρος, αλλά και να προσφέρεται για την εκτέλεση μαθηματικών πράξεων στο υλικό του υπολογιστή. Η πιο απλή κωδικοποίηση είναι η δυαδική: εδώ η (δυαδική) αναπαράσταση ενός ακεραίου αποθηκεύεται απευθείας σε μία λέξη. Ωστόσο, ο ίδιος αριθμός θα μπορούσε να κωδικοποιηθεί με πολλούς διαφορετικούς τρόπους.

Προβλήματα κωδικοποίησης Η κωδικοποίηση ενός αριθμού εισάγει κάποια προβλήματα που πρέπει να ληφθούν υπόψη κατά την εκτέλεση αριθμητικών πράξεων. Δεν δύναται να κωδικοποιηθούν όλοι οι πραγματικοί αριθμοί σε ένα ψηφιακό σύστημα με πεπερασμένο μήκος λέξης. Κατά την κωδικοποίηση δεκαδικών αριθμών, εισάγεται σφάλμα **ακρίβειας** ή σφάλμα κβαντισμού, καθώς δεν διατίθενται τόσα δεκαδικά ψηφία όσα είναι αναγκαία. Κατά την κωδικοποίηση ακεραίων αριθμών, ενώ η ακρίβεια είναι άπειρη, υπάρχει ο περιορισμός του **δυναμικού εύρους**. Δηλαδή είναι δυνατό να κωδικοποιηθούν αριθμοί που βρίσκονται σε μία περιοχή των ακεραίων αριθμών. Κατά μια έννοια, η ακρίβεια του αριθμού εξαρτάται από τον αριθμό των bit που διατίθενται για το δεκαδικό μέρος, ενώ το δυναμικό εύρος του από τα bit που διατίθενται για το ακέραιο μέρος.

Εφεξής ο όρος αναπαράσταση θα αναφέρεται σε κωδικοποιημένες δυαδικές αναπαραστάσεις. Δηλαδή το μήκος λέξης στην οποία αποθηκεύεται μία αναπαράσταση ενός δυαδικού αριθμού θα θεωρείται πεπερασμένο, με ότι αυτό συνεπάγεται. Στο κύκλωμα που υλοποιήθηκε χρησιμοποιείται το δυαδικό σύστημα και εξής αναπαραστάσεις: απροσημαστος ακεραίος (unsigned integer), προσημασμένος ακεραίος (signed integer) και (προσημασμένος ή μη) δεκαδικός σταθερής υποδιαστολής (signed/unsigned fixed point decimal), για τις οποίες ακολουθεί σύντομη περιγραφή.

4.5.1 Signed/Unsigned Integer

Ένας θετικός ακεραίος αριθμός που αναπαρίσταται ως unsigned integer εύρους N , είναι ο αριθμός (σε δυαδικό σύστημα) αποθηκευμένος απευθείας σε μία λέξη μήκους N , με το λιγότερο σημαντικό bit να γράφεται δεξιά και το περισσότερο σημαντικό bit, αριστερά. Η αξία της θέσης των ψηφίων είναι δυνάμεις του 2, από το 2^0 στη δεξιότερη θέση, έως το 2^{N-1} στην αριστερότερη. Δηλαδή ο αριθμός αναπαρίσταται ως $A = X_{N-1}X_{N-2} \dots X_0$, $X_n \in \{0, 1\}$ και η αξία του είναι

$$V_U(A) = \sum_{k=0}^{N-1} X_k 2^k$$

Ο τύπος unsigned integer συμβολίζεται ως $\langle u | N \rangle$ και το δυναμικό εύρος του είναι

$$\Delta(A) = [0, 2^N - 1]$$

Για την αναπαράσταση προσημασμένων ακεραίων αριθμών πρέπει να χρησιμοποιηθεί ένα bit για το πρόσημο. Ο πιο συνηθισμένος τρόπος, που εφαρμόζεται και στο παρόν, είναι η αναπαράσταση συμπληρώματος ως προς 2 (2's complement). Ένας αριθμός που κωδικοποιείται ως signed integer εύρους N με αυτόν τον τρόπο, αποθηκεύεται όπως ένας unsigned integer με τη διαφορά ότι το αριστερότερο bit X_{N-1} ονομάζεται bit προσήμου (sign bit) και έχει ιδιαίτερη ερμηνεία. Αν είναι '0' το πρόσημο είναι +, ενώ αν είναι '1'

το πρόσημο είναι $-$. Αν υποθεθεί ότι ο αριθμός αναπαρίσταται ως $A = X_{N-1} \dots X_0$, τότε το συμπλήρωμα ως προς 2 είναι $\bar{A} = \bar{X}_{N-1} \dots \bar{X}_0$ και υπολογίζεται με μία λογική πράξη NOT σε όλα τα bit και κατόπιν με την αριθμητική πρόσθεση '1'. Η αξία του αριθμού είναι:

$$V_S(A) = \begin{cases} \sum_{k=0}^{N-1} X_k 2^k & X_{N-1} = '0' \\ -\sum_{k=0}^{N-1} \bar{X}_k 2^k & X_{N-1} = '1' \end{cases}$$

Ο τύπος signed integer συμβολίζεται ως $\langle s | N \rangle$ και το δυναμικό εύρος του είναι

$$\Delta(A) = [-2^{N-1}, 2^{N-1} - 1]$$

4.5.2 Fixed Point

Οι βασικές αναπαραστάσεις **δεκαδικών αριθμών** σε έναν ψηφιακό υπολογιστή είναι:

- σταθερής υποδιαστολής (fixed point)
- κινητής υποδιαστολής (floating point)

Στη συνέχεια θα επεξηγηθεί μόνο η fixed point, καθώς χρησιμοποιείται στο παρόν. Η αναπαράσταση αυτή μοιάζει με την integer εκτός από το ότι υπάρχει ένα σημείο υποδιαστολής, που βρίσκεται σε σταθερή θέση σε σχέση με το δεξιότερο άκρο της αναπαράστασης του αριθμού. Ένας N -ψήφιος αριθμός με M δεκαδικά ψηφία γράφεται ως

$$B = X_{N-M-1} \dots X_0 . X_{-1} \dots X_{-M}, \quad X_n \in \{ '0', '1' \}$$

. Συνεπώς από τα N bit, τα M δεξιότερα αναπαριστούν το δεκαδικό μέρος και τα $N - M$ αριστερότερα αναπαριστούν το ακέραιο μέρος. Προφανώς είναι $N \geq M$. Για το κλασματικό μέρος, οι αξίες των θέσεων δεξιά της υποδιαστολής είναι δυνάμεις του $1/2$, ξεκινώντας από το $(1/2)^1$ στο bit ακριβώς δεξιά της υποδιαστολής, μέχρι το $(1/2)^M$ στο δεξιότερο bit. Η αξία ενός unsigned fixed point αριθμού είναι

$$V_{FU}(B) = \sum_{k=-M}^{N-M-1} X_k 2^k$$

Η αξία ενός signed fixed point με χρήση του συμπληρώματος ως προς δύο, προκύπτει με παρόμοιο τρόπο όπως για τους integer, αρκεί να εκτελεστεί η πράξη του συμπληρώματος σε όλη τη λέξη της αναπαράστασης ώστε

$$\bar{B} = \bar{X}_{N-M-1} \dots \bar{X}_0 . \bar{X}_{-1} \dots \bar{X}_{-M}$$

. Η αξία ενός signed fixed point αριθμού είναι:

$$V_{FS}(A) = \begin{cases} \sum_{k=-M}^{N-M-1} X_k 2^k & X_{N-1} = '0' \\ -\sum_{k=-M}^{N-M-1} \bar{X}_k 2^k & X_{N-1} = '1' \end{cases}$$

Ο τύπος signed fixed point συμβολίζεται ως $\langle s | N . M \rangle$ με δυναμικό εύρος

$$[-2^{N-M-1}, 2^{N-M-1} - 1]$$

Ο τύπος unsigned fixed point συμβολίζεται ως $\langle u | N . M \rangle$ με δυναμικό εύρος

$$[-2^{N-M}, 2^{N-M} - 1]$$

Είναι άξιο να σημειωθεί πως ένας (προσημασμένος) δεκαδικός αριθμός fixed point B , μπορεί να ερμηνευτεί ως ένας (προσημασμένος) ακέραιος αριθμός $I\{B\}$, που καλείται αποθηκευμένος ακέραιος (stored integer). Η αξία του αριθμού B μπορεί να εξαχθεί γνωρίζοντας τον $I\{B\}$ και τις τιμές των M, N . Ουσιαστικά η κωδικοποίηση ενός fixed point δεν διαφέρει από αυτή ενός integer. Αυτό που διαφέρει είναι η

ερμηνεία της κωδικοποίησης κατά την αποτίμησή της τιμής της. Κατ' επέκταση, σε έναν ψηφιακό υπολογιστή, δεν έχει καμία διαφορά η εκτέλεση των βασικών πράξεων (+, -, ×, /) πάνω σε έναν fixed point από όταν αυτές εκτελούνται πάνω σε έναν integer αριθμό. Με λίγα λόγια μπορεί να χρησιμοποιηθούν οι ίδιες δομές hardware για την εκτέλεση πράξεων μεταξύ fixed point με αυτές που χρησιμοποιούνται για τους integer. Είναι θέμα ερμηνείας των integer αριθμών που μετέχουν σε μία πράξη ως τον κατάλληλο τύπο fixed point.

4.5.3 Στρογγυλοποίηση

Κατά την αναπαράσταση ενός δεκαδικού αριθμού σε fixed point, είναι πιθανό να εισαχθεί σφάλμα από τη στρογγυλοποίηση. Η τελευταία είναι μία διαδικασία που πρέπει να εκτελεστεί πάνω σε έναν αριθμό εάν αυτός δεν δύναται να αναπαρασταθεί δεδομένου ενός τύπου fixed point. Βασικοί τρόποι στρογγυλοποίησης είναι η στρογγυλοποίηση στον ελάχιστο (floor), στον μέγιστο (ceil) και στον κοντινότερο ακέραιο (round).

Αν $\epsilon = a - \bar{a}$ είναι το σφάλμα λόγω στρογγυλοποίησης του δεκαδικού αριθμού a , όταν αυτός αναπαρίσταται σε τύπο fixed point $\langle N.M \rangle$ ως \bar{a} , τότε για τις floor και ceil το μέγιστο σφάλμα είναι $\max(\epsilon) = 2^{-M}$, ενώ για τη round είναι $\max(\epsilon) = \frac{2^{-M}}{2}$.⁴

Συνήθως στα ψηφιακά κυκλώματα η floor είναι ταχύτερη λειτουργία μιας και υλοποιείται με αποκοπή των περισσευόμενων ψηφίων (truncation), ενώ η round χρειάζεται την ύπαρξη ενός μικρού αθροιστή. Για αυτό το λόγο η κύρια μέθοδος στρογγυλοποίησης στο παρόν θα είναι η floor εκτός εάν αναφερθεί κάτι διαφορετικό.

4.5.4 Υπερχείλιση - Κορεσμός

Από την άλλη κατά την αναπαράσταση ενός ακέραιου αριθμού με έναν τύπο integer ή fixed point, η ακρίβεια είναι άπειρη, αλλά δεν δύναται να αναπαρασταθούν όλοι οι ακέραιοι λόγω του περιορισμού του δυναμικού εύρους. Αν ένας αριθμός δε «χωράει» σε έναν τύπο, συμβαίνει υπερχείλιση προς τα πάνω (overflow) ή προς τα κάτω (underflow). Οι κύριοι μετασχηματισμοί που μπορούν να γίνουν ώστε να αντιμετωπιστεί μία υπερχείλιση είναι, ο κορεσμός (saturation), η αναδίπλωση (rewind) και η αποκοπή (truncation).

Η μέθοδος που παρουσιάζει το μικρότερο σφάλμα είναι ο κορεσμός, όπου ο επιλέγεται ο μέγιστος αριθμός που αναπαρίσταται και είναι πλησιέστερος στην πραγματική τιμή του αριθμού. Για την υλοποίηση σε hardware πρέπει να υλοποιηθεί ένα μικρό κύκλωμα ελέγχου υπερχείλισης. Σε συστήματα DSP, όπως το παρόν, όταν ένα σήμα υποστεί κορεσμό λόγω μεγάλου πλάτους, ψαλιδίζονται οι κορυφές του, ένα φαινόμενο γνωστό ως clipping. Τα αποτελέσματα του clipping μπορεί να έχουν μεγάλη ή αμελητέα επίπτωση ανάλογα με το μέγεθος του σφάλματος, αλλά και την ευαισθησία της εφαρμογής. Για παράδειγμα αν το σήμα πρόκειται να μετατραπεί σε αναλογικό, το clipping μπορεί να αποβεί αρκετά καταστροφικό, καθώς εισάγει αρμονικές στο σήμα εξόδου.

Στη μέθοδο αναδίπλωσης, όταν ξεπεραστεί το ένα όριο, έστω το μέγιστο, ο αριθμός γυρίζει στο άλλο όριο, έστω το ελάχιστο συν το περίσσειμα. Αυτή η συμπεριφορά είναι πολύ χρήσιμη σε μετρητές με full adders, καθώς τους κάνει να μηδενίζουν αυτόματα χωρίς επιπλέον κύκλωμα. Δεν έχει κάποιο ιδιαίτερο νόημα στην αναπαράσταση σημάτων.

Στη μέθοδο της αποκοπής αγνοούνται τα περισσευόμενα MSB του αριθμού οδηγώντας σε μία τυχαία τιμή εντός του δυναμικού εύρους. Αυτή η συμπεριφορά πρέπει να αποφεύγεται καθώς τα αποτελέσματα δεν έχουν κάποιο ιδιαίτερο νόημα. Επιπλέον όταν αναπαρίσταται ένα σήμα, η παραμόρφωση που μπορεί να εισαχθεί είναι τυχαία. Ωστόσο η αποκοπή παρουσιάζει το πλεονέκτημα της πολύ απλής υλοποίησης και επιλέγεται κατά τον σχεδιασμό όταν υπάρχει βεβαιότητα ως προς το όριο του αριθμού που πρόκειται

⁴Σε μία fixed point αναπαράσταση σε ένα άλλο σύστημα αριθμητικής με βάση b , τα μέγιστα σφάλματα στρογγυλοποίησης για τις ceil, floor είναι $\max(\epsilon) = b^{-M}$, ενώ για τη round είναι $\max(\epsilon) = b^{-M}/2$.

να αναπαρασταθεί.

Για τον χειρισμό της υπερχειλίσης των σημάτων στο παρόν σύστημα χρησιμοποιείται τόσο κορεσμός, όσο και αποκοπή.

Ο τύπος ενός fixed point αριθμού που χρησιμοποιείται σε ένα σύστημα για την εκτέλεση μιας πράξης επιλέγεται μετά από ανάλυση δυναμικού εύρους και ανάλυση ακρίβειας.

Ένα παράδειγμα

Τα ακόλουθα παραδείγματα είναι γραμμένα σε δεκαδικό σύστημα αριθμητικής για λόγους κατανόησης. Ακριβώς τα ίδια ισχύουν και στο δυαδικό σύστημα.

Ο αριθμός $a = 7.234$ αναπαρίσταται επ' ακριβώς με fixed point $\langle u | 4 . 3 \rangle$ και αποθηκεύεται ως

$$X_0.X_{-1}X_{-2}X_{-3} \quad \bar{a} = 7.234$$

. Το δυναμικό εύρος του τύπου είναι $[0, 10)$. Ο αριθμός $b = 5.2$ αναπαρίσταται επίσης επακριβώς αρκεί να προστεθούν μηδενικά στα δεξιά του, ώστε η υποδιαστολή να μετακινηθεί στο ίδιο σημείο με πριν, $\bar{b} = 5.200$. Ωστόσο ο αριθμός $c = 9.4157$ δεν αναπαρίσταται επ' ακριβώς. Μία πιθανή αναπαράσταση, με το μετασχηματισμό floor, είναι $\bar{c} = 9.415$, με σφάλμα ακρίβειας $\epsilon = 0.0007 < 10^{-3} = 0.001$. Αν γίνει ο μετασχηματισμός round, είναι $\hat{c} = 9.416$, με σφάλμα $\hat{\epsilon} = 0.0003 < \frac{10^{-3}}{2} = 0.0005$. Τέλος, ο αριθμός $d = 12.43$ δεν χωράει στο δυναμικό εύρος και μπορεί να αναπαρασταθεί με το μετασχηματισμό saturation ως $\bar{d} = 9.999$ με σφάλμα κορεσμού $\epsilon = 2.431$. Αν γίνει ο μετασχηματισμός truncate, ο αριθμός αναπαρίσταται ως $\hat{d} = 2.430$ με σφάλμα truncation $\hat{\epsilon} = 10.000$.

4.6 Time Synchroniser [TSY]

Ο TSY πραγματοποιεί συγχρονισμό στο χρόνο και υπολογίζει την fine εκτίμηση ολίσθησης συχνότητας. Υλοποιεί τον αλγόριθμο S&C όπως περιγράφεται αναλυτικά στην § 2.2.3. Εισέρχεται στον TSY μία σταθερή ροή samples και εξέρχονται ομαδοποιημένα samples που αποτελούν OFDM σύμβολα, μετά την ανίχνευση της αρχής του frame (SOF). Ταυτόχρονα παρέχεται η fine εκτίμηση διαφοράς φάσης, η οποία χρησιμοποιείται από το επόμενο module, τον FCO, για τον συγχρονισμό συχνότητας.

4.6.1 Η Top-Level Όψη

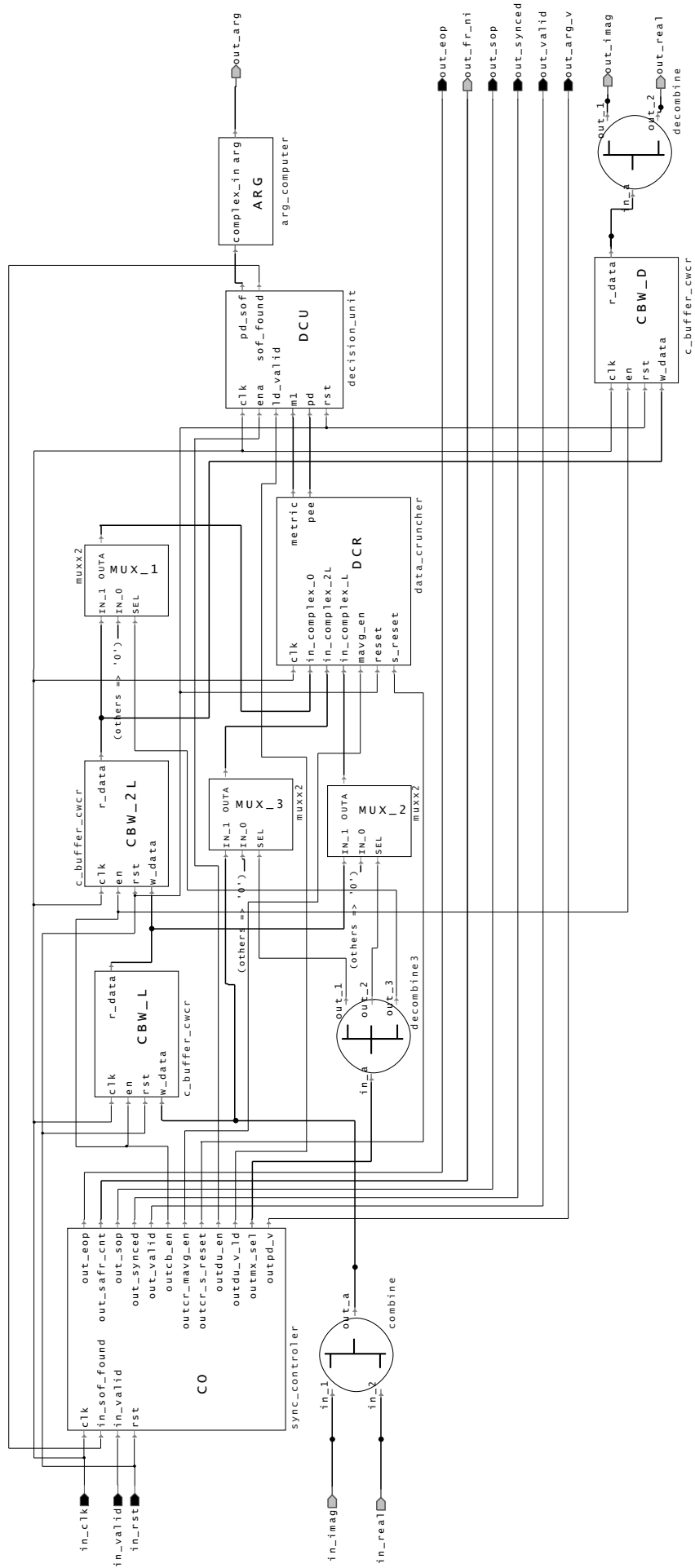
Η εσωτερική αρχιτεκτονική του TSY περιλαμβάνει τα εξής module:

- Controller (TSY.CO)
- 3 x CBuffer Constant Writing (TSY.CBW_L, TSY.CBW_2L, TSY.CBW_D)
- Data Cruncher (TSY.DCR)
- Arg Computer (TSY.ARG)
- Decision Unit (TSY.DCU)

και απεικονίζεται στο Σχήμα 4.4.

Συνοπτική Περιγραφή Υποσυστημάτων

Ο CO είναι ο γενικός ελεγκτής όλων των υποσυστημάτων που αποτελούν τον TSY. Διατηρεί την κατάσταση του κυκλώματος (data, sync mode) και φροντίζει για την ομαλή μετάβαση των καταστάσεων. Αρχικοποιεί τον DCR και τον DCU. Επιπλέον οδηγεί τα σήματα του Avalon στην έξοδο του TSY.



Σχήμα 4.4: TSY Top-Level διάγραμμα

Οι κυκλικόι buffer (CBW_L, CBW_2L, CBW_D) αποτελούν όμοιες δομές στις οποίες τα δεδομένα γράφονται και διαβάζονται κυκλικά με συνεχή τρόπο, έτσι ώστε τα νεότερα δεδομένα να γράφονται πάνω στα παλαιότερα. Η ανάγνωση γίνεται από τα παλαιότερα δεδομένα λίγο πριν αυτά αντιγραφούν. Οι 3 buffer είναι συνδεδεμένοι σε αλυσίδα, αποτελώντας ένα μεγάλο κυκλικό buffer, με 3 taps εξόδου.

Ο DCR είναι ο υπολογιστής των μετρικών $P(d)$, $M_A(d)$. Λόγω των «σφιχτών» χρονικών απαιτήσεων, ο DCR έχει σχεδιαστεί με συνδυαστική λογική, έτσι ώστε να υπολογίζει ένα sample των μετρικών ανά clock, εισάγοντας το μικρότερο δυνατό latency. Πριν από τις εισόδους του DCR, παρεμβάλλονται MUX, οι οποίοι λειτουργούν σαν «μεμβράνες» που ελέγχουν τα samples που εισέρχονται στον DCR. Αυτό είναι αναγκαίο καθώς μέσα στον DCR υπάρχουν συσσωρευτές και μνήμη, συνεπώς θα πρέπει να παρθεί ειδική μέριμνα για την διατήρηση της συνέπειας της κατάστασης τους.

Ο DU εκτελεί τον αλγόριθμο αναζήτησης μεγίστου της μετρικής $M_A(d)$. Όταν βρεθεί το μέγιστο σηκώνει το σήμα `sof_found` και ταυτόχρονα τροφοδοτεί την $P(d)$ που αντιστοιχεί στο μέγιστο σημείο, στον ARG.

Ο ARG είναι ένας υπολογιστής του ορίσματος ενός μιγαδικού αριθμού. Όπως και ο DCR, έχει υλοποιηθεί με συνδυαστική λογική, επιτυγχάνοντας τον υπολογισμό του ορίσματος με το ελάχιστο latency.

Υπάρχουν δύο βασικές καταστάσεις λειτουργίας για τον TSY:

- sync mode
- data mode

Στο sync mode, γίνεται αναζήτηση για την αρχή του frame (Start Of Frame - SOF) και δεν εξέρχονται δεδομένα, με το `out_valid` να είναι κάτω ασκώντας `frontpressure` στα downstream modules. Μόλις βρεθεί το SOF, η κατάσταση αλλάζει σε data mode και τα συγχρονισμένα samples εξέρχονται από τις θύρες `out_real` και `out_imag`, με το `out_valid` να έχει σηκωθεί σηματοδοτώντας μεταφορά δεδομένων. Επίσης στην έξοδο `out_arg` εξέρχεται η εκτίμηση φάσης, με το `out_arg_v` να σηματοδοτεί την εγκυρότητά της. Ο TSY παραμένει σε data mode μέχρι το πέρας του frame, οπότε μεταβαίνει πάλι σε sync mode και ο κύκλος επαναλαμβάνεται.

Όταν το κύκλωμα βρίσκεται σε sync mode, τα δεδομένα ρέουν μέσω των CBW_L, CBW_2L στον DCR. Εκεί υπολογίζεται η μετρική $M_A(d)$ και η $P(d)$. Κατόπιν ο DU αναζητά το μέγιστο της $M_A(d)$ και όταν το βρει, ενημερώνει τον CO μέσω του σήματος `sof_found`. Τότε, το κύκλωμα γυρίζει σε data mode και τα δεδομένα ρέουν μέσω του CBW_D στην έξοδο, ενώ ο CO παράγει τα σήματα `Analog`, που ελέγχουν την ροή των δεδομένων στα downstream modules.

Μόλις ο TSY εξέλθει από την κατάσταση sync, ο CO φροντίζει για τον μηδενισμό της εσωτερικής κατάστασης του DCR, ώστε να είναι έτοιμος για τον επόμενο sync κύκλο. Ταυτόχρονα εμποδίζει νέα samples να εισέλθουν στον DCR, μέσω του μηχανισμού των μεμβρανών. Αντί αυτών εισέρχονται μηδενικά τα οποία δεν αλλοιώνουν την εσωτερική κατάσταση. Επίσης ο CO, απενεργοποιεί τον DU ώστε να μην είναι δυνατό να στείλει κάποιο ψευδές συμβάν `sof_found`.

Λίγο πριν την έναρξη του sync κύκλου⁵ οι μεμβράνες ανοίγουν η μία μετά την άλλη, σε κατάλληλα χρονικά διαστήματα, αρχικοποιώντας την εσωτερική κατάσταση του DCR. Αυτό γίνεται για το φόρτωμα των accumulator του DCR με έγκυρα δεδομένα. Στη συνέχεια ο DU, ενεργοποιείται ακριβώς τη χρονική στιγμή που εξέρχεται η πρώτη έγκυρη μετρική από τον DCR και ο CO αναμένει το συμβάν `sof_found`.

Τα Επιμέρους Υποσυστήματα

4.6.2 Κυκλικόι Buffer [TSY.CBW_*]

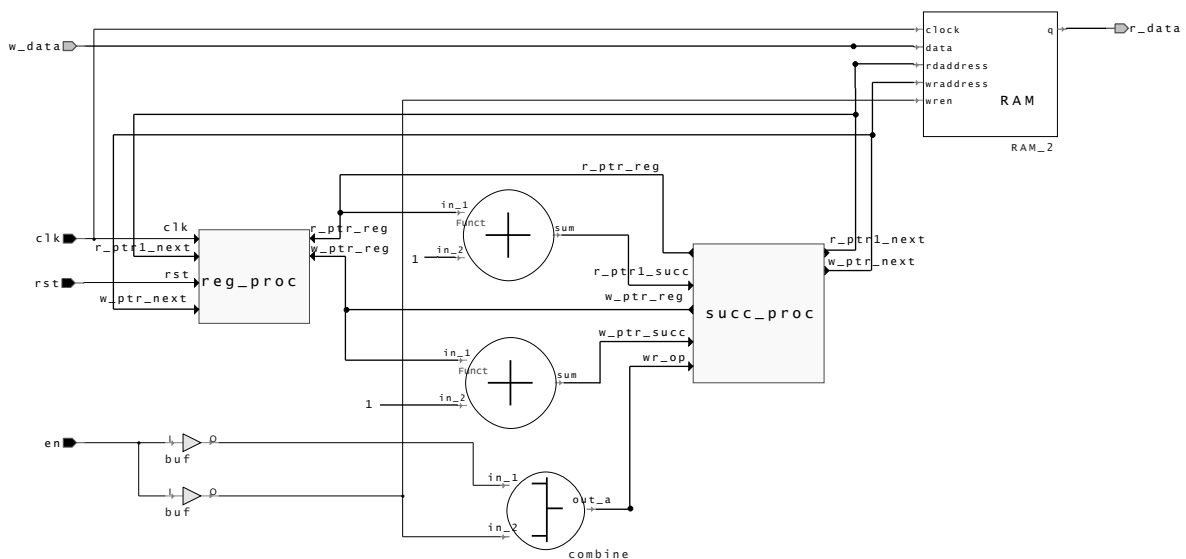
```
component c_buffer_cwcr is
```

⁵Εκτός από την πρώτη εκτέλεση του κυκλώματος, όπου οι μεμβράνες είναι ήδη ανοιχτές

```

generic (
  DATA_WIDTH : natural :=16; -- number of bits
  ADDR_WIDTH  : natural :=8;  -- number of address bits MAX 8 | BUFF_LEN : natural :=
    2**ADDR_WIDTH;
  READ_OFFSET : integer      -- range -BUFF_LEN ~ -2 constant
);
port (
  clk      : in std_logic;
  rst      : in std_logic;
  en       : in std_logic;
  w_data   : in std_logic_vector (DATA_WIDTH-1 downto 0);
  r_data   : out std_logic_vector (DATA_WIDTH-1 downto 0)
  -- input offset set to output delay 1
);
end component;

```



Σχήμα 4.5: TSY.CBW_* Top-Level διάγραμμα

Ο CBW υλοποιείται με τη χρήση μίας μνήμης RAM ως βασική δομή αποθήκευσης και ενός λογικού κυκλώματος που αποτελεί την «κυκλική» διεπαφή της μνήμης. Οι βασική παράμετρος (generic) του κυκλώματος είναι η θέση ανάγνωσης σχετικά με την τρέχουσα θέση εγγραφής `READ_OFFSET`. Το `READOFFSET` μπορεί να πάρει τιμές `[-BUFFER_LENGTH ... -2]`. Οι 3 κυκλικοί buffer που αναφέρθηκαν, συν ακόμα έναν που υπάρχει μέσα στον DCR, δεν είναι παρά διαφορετικά στιγμιότυπα του ίδιου κυκλώματος με διαφορετικά `READ_OFFSET`. Το top-level διάγραμμα του CBW, παρουσιάζεται στο σχήμα 4.5.

Το κύκλωμα περιλαμβάνει δυο δείκτες `r_ptr_next`, `w_ptr_next` για την διεύθυνση ανάγνωσης και εγγραφής στη μνήμη, αντίστοιχα. Οι τρέχουσες τιμές των δεικτών καταχωρούνται στο `reg_proc` στα σήματα `w_ptr_reg` και `r_ptr_reg`. Με τη βοήθεια των registered σημάτων παράγονται οι επόμενες τιμές των δεικτών, στα σήματα `w_ptr_succ` και `r_ptr_succ`. Το `succ_proc` αναλαμβάνει να συνδέσει τους δείκτες `*_succ` με τους αντίστοιχους `*_next` όταν το σήμα `en` είναι '1'. Σε αντίθετη περίπτωση οι μετρητές παγώνουν στην προηγούμενη τιμή τους, που είναι αποθηκευμένη στα `*_reg` σήματα.

Οι δείκτες αυξάνονται σε κάθε clock μέσω δύο adders, με τρόπο που αναδιπλώνονται όταν φτάσουν στο τέλος του buffer. Ο `w_ptr_reg`, αρχικοποιείται στο 0, ενώ ο `r_ptr_reg`, αρχικοποιείται στην τιμή που αντιστοιχεί στο `READ_OFFSET`, με τρόπο τέτοιο ώστε στην έξοδο να παρουσιάζεται το δεδομένο που είχε γραφτεί πριν από `READ_OFFSET` samples. Η διαδικασία ανάγνωσης και εγγραφής μπορεί να σταματήσει, αν πέσει το σήμα `en`.

RAM

```

component ram_2cw_mod IS
  GENERIC
  (
    ADDR_WIDTH : natural := 8 --address width (4~16)
  );
  PORT
  (
    clock      : IN STD_LOGIC := '1';
    data       : IN STD_LOGIC_VECTOR (31 DOWNTO 0);
    rdaddress  : IN STD_LOGIC_VECTOR (ADDR_WIDTH-1 DOWNTO 0);
    wraddress  : IN STD_LOGIC_VECTOR (ADDR_WIDTH-1 DOWNTO 0);
    wren       : IN STD_LOGIC := '0';
    q          : OUT STD_LOGIC_VECTOR (31 DOWNTO 0)
  );
end component;

```

Τύπος	Altsyncram • Dual Port
Εύρος θύρας διεύθυνσης	GENERIC
Εύρος θύρας διεύθυνσης εισόδου/εξόδου CBW_L	LOG2_SZ_USYM bit
Εύρος θύρας διεύθυνσης CBW_2L	LOG2_SZ_USYM bit
Εύρος θύρας διεύθυνσης CBW_D	LOG2_SZ_USYM+1 bit
Εύρος θύρας διεύθυνσης CBW_M	11 bit
Εύρος θύρας εισόδου/εξόδου	32 bit

Πίνακας 4.7: Τεχνικά Χαρακτηριστικά Altera RAM (CBW)

Η υφιστάμενη RAM είναι «τυποποιημένη» Altera RAM δύο θυρών, μίας εισόδου και μιας εξόδου. Ουσιαστικά πρόκειται για ελαφρώς τροποποιημένη εκδοχή, καθώς το εύρος των διεύθυνσεων εισόδου και εξόδου είναι generic, δηλαδή κυκλωματική παράμετρος. Οι τιμές της παραμέτρου για τα τρία στιγμιότυπα της RAM (που υπάρχει στους 3 CBW) αναγράφονται στον πίνακα στον πίνακα 4.7. Η παράμετρος LOG2_SZ_USYM ορίζεται στις εξωτερικές παραμέτρους του κυκλώματος και διανέμεται από το μηχανισμό των *Asset Dispatching*. Η RAM διαθέτει επιπλέον, θύρα ενεργοποίησης εγγραφής *wren*. Τα χαρακτηριστικά της συνοψίζονται στον πίνακα 4.7.

Τα στιγμιότυπα του CBW

Όπως αναφέρθηκε, οι 3 κυκλικοί buffer είναι στιγμιότυπα του κυκλώματος CBW.⁶ Τα READ_OFFSET για κάθε buffer είναι:

$$CBW_L \rightarrow -L \quad CBW_2L \rightarrow -L \quad CBW_D \rightarrow -(M_{span} + S_{offset} + 2 + N_{CP})$$

Με τη θεώρηση του συστήματος των buffer ως έναν ενιαίο κυκλικό buffer μίας εισόδου και 3 εξόδων, τα read offsets είναι: $-L$, $-2L$, $-(2L + M_{span} + S_{offset} + 2 + N_{CP})$. Οι δύο πρώτες έξοδοι χρησιμοποιούνται για τον υπολογισμό του M_A και συνδέονται στον DCR. Τα offset προκύπτουν από την υλοποίηση του DCR και θα εξηγηθούν στο αντίστοιχο χωρίο. Η τρίτη έξοδος, αποτελεί την έξοδο δεδομένων του κυκλώματος *out_real*, *out_imag*. Η τιμή του offset προκύπτει από την αρχιτεκτονική του συστήματος και την προδιαγραφή για επιλογή του βέλτιστου σημείου συγχρονισμού.

Συγκεκριμένα, το M_{span} προέρχεται από τον αλγόριθμο αναζήτησης μεγίστου του DU, καθώς όπως θα εξηγηθεί και παρακάτω, όταν συμβαίνει το *sof_found* αναφέρεται στην μετρική που είχε εισέλθει στον

⁶Υπάρχει και ένα τέταρτο στιγμιότυπο, ο CBW_M, που βρίσκεται μέσα στον TSY.DCR.MVG. Θα περιγραφεί στο αντίστοιχο εδάφιο.

DU, M_{span} samples νωρίτερα. Το $2L$ προέρχεται από το ότι η μετρική $M(d)$ υπολογίζεται πάνω στα samples $r(m-2L) \dots r(m-1)$. Το N_{CP} προέρχεται από την απαίτηση, το frame να αρχίζει από το πρώτο OFDM σύμβολο και όχι το πρώτο ωφέλιμο σύμβολο, εφόσον η αφαίρεση της κυκλικής επέκτασης γίνεται αργότερα, από άλλο module στο downstream. Αν ληφθεί υπόψη πως η $M_A(d)$ μεγιστοποιείται στην αρχή του ωφέλιμου ofdm συμβόλου, είναι προφανές πως πρέπει να εξαχθούν N_{CP} samples νωρίτερα από το σημείο μεγιστοποίησης της $M_A(d)$. Το S_{offset} είναι μία παράμετρος που εξασφαλίζει ότι το σημείο έναρξης του DFT για την αποδιαμόρφωση, θα είναι στο μέσον της περιοχής ασφαλείας, παρέχοντας ανεκτικότητα σε σφάλματα συγχρονισμού.⁷ Το 2 είναι μία «μαγική τιμή» ώστε το σύστημα να εκπληρώνει τους χρονικούς του περιορισμούς, προερχόμενη από εσωτερική καθυστέρηση που υπάρχει στον CO.

4.6.3 Metric Computer [TSY.DCR]

```

component data_cruncher is
  generic (
    HALF_LOG2_SCALE : integer := 5;
    CMPLX_WIDTH     : natural  := 32;
    ACC_WIDTH       : natural  := 32;
    SZ_PLATEAU     : natural  := 30
  );
  port (
    clk, reset : in std_logic;
    in_complex_0 : in std_logic_vector(CMPLX_WIDTH-1 downto 0);
    in_complex_L : in std_logic_vector(CMPLX_WIDTH-1 downto 0);
    in_complex_2L : in std_logic_vector(CMPLX_WIDTH-1 downto 0);
    mavg_en      : in std_logic;
    s_reset      : in std_logic;
    metric       : out std_logic_vector(2*HALF_LOG2_SCALE downto 0); -- scaled metric + 1 bit for
      guard
    pee         : out std_logic_vector(2*ACC_WIDTH-1 downto 0) --complex
  );
end component;

```

Η υπολογιστής μετρικής υπολογίζει τη μετρική $M_A(d)$ μέσω του αναδρομικού τύπου (2.4). Για να γίνει αυτό, υπολογίζονται τα βοηθητικά μεγέθη $M_f(d)$ (2.1), $P(d)$ (2.2), $R_f(d)$ (2.3). Για την εκτέλεση των σύνθετων πράξεων, αυτές αναλύονται σε μικρότερες παραστάσεις, ώστε να υπάρχει μεγαλύτερος έλεγχος στην ακρίβεια της κάθε πράξης. Η μετρική $M_A(d)$ υπολογίζεται από την $M_f(d)$ μέσω ενός υπολογιστή κινούμενου μέσου όρου (Moving Average Computer, TSY.DCR.MVG).

Για τον υπολογισμό της $P(d)$ υπολογίζεται το πραγματικό και το φανταστικό μέρος της ξεχωριστά. Οι μαθηματικές εκφράσεις που υπολογίζονται για το $P(d)$ είναι:

$$\text{re}_{\tilde{P}_{d+1}} = \text{re}_{\tilde{P}_d} + \frac{\text{re}_{r-L} \times [\text{re}_{r_0} - \text{re}_{r-2L}] + \text{im}_{r-L} \times [\text{im}_{r_0} - \text{im}_{r-2L}]}{\text{scale}_P}$$

$$\text{im}_{\tilde{P}_{d+1}} = \text{im}_{\tilde{P}_d} + \frac{\text{re}_{r-L} \times [\text{im}_{r_0} + \text{im}_{r-2L}] - \text{im}_{r-L} \times [\text{re}_{r_0} + \text{re}_{r-2L}]}{\text{scale}_P}$$

όπου r_x το sample $r(x)$, r_0 το τρέχων sample στην είσοδο του TSY, \tilde{P}_{d+1} η τρέχουσα τιμή της μετρικής και \tilde{P}_d ο accumulator, δηλαδή η προηγούμενη τιμή της. Χρησιμοποιείται ένας μιγαδικός accumulator, για το πραγματικό και το φανταστικό μέρος της $P(d)$.

Η πρόσθεση των μερικών γινομένων στον accumulator, γίνεται με saturated αριθμητική για την αποφυγή αναδίπλωσής του. Επίσης, αντί της $P(d)$ υπολογίζεται η $\tilde{P}(d) = P(d)/\text{scale}_P$. Ο συντελεστής κλιμάκωσης scale_P έχει εισαχθεί για την προστασία του accumulator από παρατεταμένο κορεσμό. Αυτό μπορεί να συμβεί γιατί οι τιμές της $P(d)$ φτάνουν σε πολύ υψηλά επίπεδα. Στην πραγματικότητα ο παράγοντας κλιμάκωσης είναι $\text{scale}_P = 2^{\log_2 \text{scale}_P}$, ώστε να υλοποιείται με μία απλή ολίσθηση κατά $\log_2 \text{scale}_P$ θέσεις προς τα δεξιά. Η πολυπλοκότητα $\forall d$ είναι :

⁷Βλ. § 2.2.3

8xADD, 4xMULT, 2xSHIFT

Πραγματοποιούνται επίσης ($\forall d$) και δύο προσπελάσεις ανάγνωσης στη μνήμη (2xLOAD).

Για τον υπολογισμό της $R_f(d)$ χρησιμοποιείται ο τύπος:

$$\tilde{R}_{f_{d+1}} = \tilde{R}_{f_d} + \frac{[\text{re}_{r_0} \times \text{re}_{r_0} + \text{im}_{r_0} \times \text{im}_{r_0}] - [\text{re}_{r_{-2L}} \times \text{re}_{r_{-2L}} + \text{im}_{r_{-2L}} \times \text{im}_{r_{-2L}}]}{2 \text{scale}_{R_f}}$$

Οι πράξεις γίνονται παρόμοια με την $P(d)$, δηλαδή με saturated αριθμητική και με τη χρήση συντελεστή κλιμάκωσης scale_{R_f} . Ισχύει, επίσης, $\text{scale}_{R_f} = 2^{\log_2 \text{scale}_{R_f}}$.⁸ Εν τέλει, υπολογίζεται η μετρική $\tilde{R}_f(d) = R_f(d)/\text{scale}_{R_f}$. Η πολυπλοκότητα είναι $\forall d$:

4xADD, 4xMULT, 1xSHIFT, 1xLOAD

Στη συνέχεια υπολογίζονται τα τετράγωνα των παραπάνω μετρικών:

$$\tilde{P}_{d+1}^2 = [\text{re}_{\tilde{P}_{d+1}} \times \text{re}_{\tilde{P}_{d+1}}] + [\text{im}_{\tilde{P}_{d+1}} \times \text{im}_{\tilde{P}_{d+1}}]$$

$$\tilde{R}_{f_{d+1}}^2 = [\tilde{R}_{f_{d+1}} \times \tilde{R}_{f_{d+1}}]$$

Η πολυπλοκότητα είναι $\forall d$: 1xADD, 3xMULT.

Ακολουθεί ο υπολογισμός της μετρικής $M_f(d)$, που περιλαμβάνει έναν διαιρέτη:

$$\tilde{M}_{f_{d+1}} = \frac{\tilde{P}_{d+1}^2}{\tilde{R}_{f_{d+1}}^2}$$

Η μετρική $\tilde{M}_{f_{d+1}}$ διαφέρει από τη μετρική που υπολογίζεται στην (2.1) κατά έναν συντελεστή κλιμάκωσης. Η σχέση με την $M_f(d)$ είναι

$$\tilde{M}_f(d) = M_f(d) \left(\frac{\text{scale}_{R_f}}{\text{scale}_P} \right)^2 = M_f(d) 2^{2(\log_2 \text{scale}_{R_f} - \log_2 \text{scale}_P)}$$

Η πολυπλοκότητα είναι $\forall d$: 1xDIV.

Αριθμητική

Χρησιμοποιείται integer αριθμητική σε όλες τις μετρικές του DCR. Καθώς ισχύει $\max_{\forall d} (M_A(d)) = 1$ και για το λόγο ότι η τιμή της $M_A(d)$ δεν μετέχει σε περαιτέρω υπολογισμούς, υπολογίζεται η

$$\tilde{M}_A(d) = \text{floor} \left(M_A(d+1) 2^{(\log_2 \text{scale}_{M_A})} \right)$$

Δηλαδή ένα πολλαπλάσιο της τιμής της μετρικής, στρογγυλοποιημένο στον μικρότερο ακέραιο. Είναι $\max_{\forall d} (\tilde{M}_A(d)) = 2^{(\log_2 \text{scale}_{M_A})}$. Συνεπώς η αναπαράσταση της $\tilde{M}_{A_{d+1}}$ μπορεί να γίνει με unsigned integer $\langle u | \log_2 \text{scale}_{M_A} \rangle$. Στην πραγματικότητα το εύρος είναι $\log_2 \text{scale}_{M_A} + 1$, για την παροχή μεγαλύτερου δυναμικού εύρους σε περιπτώσεις όπου η μετρική ξεπεράσει το μέγιστο λόγω κάποιας ανωμαλίας στις πράξεις. Η τιμή που έχει επιλεγεί είναι $\log_2 \text{scale}_{M_A} = 10$, συνεπώς ο τύπος της $\tilde{M}(d)$ είναι $\langle u | 11 \rangle$.

Για τις ενδιάμεσες τιμές, λαμβάνοντας υπόψη την αναπαράσταση των samples εισόδου σε $\langle s | 16 \rangle$, επιλέγεται για τους accumulator $\tilde{P}_d, \tilde{R}_{f_d}$ που αποτελούνται από αθροίσματα γινομένων μεταξύ samples, τα 32 bit. Αυτή η τιμή, γενικά δεν είναι επαρκής για την αναπαράσταση των τιμών των μετρικών που απαντώνται

⁸Το 2 στον παρονομαστή προέρχεται από τον αναδρομικό τύπο και δεν έχει να κάνει με τον παράγοντα κλιμάκωσης

στην πράξη. Για αυτό εισάγεται ο συντελεστής προστασίας παρατεταμένου κορεσμού. Το μέγεθος των τιμών που μπορεί να πάρουν οι μετρικές P και R_f εξαρτάται από τη μέση ισχύ του σήματος εισόδου, καθώς και από τη μεταξύ τους σχέση. Εξάλλου, είναι αναμενόμενο η P να μεγιστοποιείται κατά την ύπαρξη του πλάτος. Η R_f είναι γενικά σε υψηλά επίπεδα κατά την ύπαρξη σήματος καθώς υπολογίζει ενέργεια. Χωρίς την ύπαρξη των συντελεστών προστασίας, οι μετρικές θα βρίσκονταν για μεγάλο χρονικό διάστημα σε κορεσμό, αλλοιώνοντας κατά πολύ την τιμή της M_A .

Καθοριστικό ρόλο στην επιλογή της ακρίβειας των 32 bit για τους accumulator έπαιξε το μέγιστο επιτρεπόμενο εύρος διαιρέτη από το εργαλείο σύνθεσης, που είναι στα 64 bit. Κατ' επέκταση οι μετρικές $\tilde{R}_{f_{d+1}}$ και \tilde{P}_{d+1} πρέπει να έχουν μέγιστο εύρος 32 bit, ώστε τα τετράγώνά τους να είναι 64 bit.

Moving average computer [TSY.DCR.MVG]

```
component mavg_comp is
  generic(
    WINDOW      : natural;
    DATA_WIDTH : natural
  );
  port(
    clk      : in std_logic;
    reset    : in std_logic;
    s_reset  : in std_logic;
    mux_sel  : in std_logic;
    data_in  : in std_logic_vector(DATA_WIDTH-1 downto 0);
    avg      : out std_logic_vector(DATA_WIDTH-1 downto 0)
  );
end component;
```

Η $\tilde{M}_f(d)$ τροφοδοτείται στον υπολογιστή κινούμενου μέσου όρου MVG, στην έξοδο του οποίου υπολογίζεται η τελική μετρική $\tilde{M}_A(d)$. Ο κινούμενος μέσος όρος, υπολογίζει τον μέσω όρο των πρόσφατων samples που βρίσκονται σε ένα παράθυρο μήκους $N_{CP} + 1$.

Η αναλυτικές εκφράσεις που υπολογίζονται είναι:

$$\tilde{M}_{B_{d+1}} = \tilde{M}_{B_d} + \left(\tilde{M}_{f_{d+1}} - \tilde{M}_{f_{d-N_{CP}}} \right)$$

$$\tilde{M}_{A_{d+1}} = \frac{\tilde{M}_{B_{d+1}}}{N_{CP} + 1}$$

Η πολυπλοκότητα ανέρχεται σε :

$$1 \times \text{DIV}, 2 \times \text{ADD} \text{ και } 1 \times \text{LOAD}$$

ενώ το I/O latency είναι 0.

Στο κύκλωμα υπάρχει ένας accumulator $\tilde{M}_{B_{d+1}}$ που υπολογίζει το κινούμενο άθροισμα των πιο πρόσφατων samples σε διάστημα ίσο με το παράθυρο παρατήρησης. Η πρόσθεση γίνεται με saturated αριθμητική. Κατόπιν το άθροισμα διαιρείται με το μήκος του παραθύρου για τον υπολογισμό του μέσου όρου. Ο λόγος για τον οποίο η διαίρεση δεν μπήκε στον αναδρομικό τύπο αλλά μετά από αυτόν, είναι ότι τα μερικά αθροίσματα της μετρικής δεν αναμένεται να είναι πολύ μεγάλα, συνεπώς μία διαίρεση πριν αυτά προστεθούν στον accumulator θα αύξανε κατά πολύ τα σφάλματα στρογγυλοποίησης.

Καθώς τα samples εισόδου $\tilde{M}_{f_{d+1}}$ είναι τύπου $\langle u | 11 \rangle$ για τον accumulator επιλέχθηκε τύπος $\langle s | 19 \rangle$ ώστε να δίνει ένα επαρκές περιθώριο ασφαλείας. Στην έξοδο μετά τον διαιρέτη το σήμα αποθηκεύεται σε $\langle u | 11 \rangle$, δηλαδή σε τύπο όμοιο με αυτόν της εισόδου.

Για την παροχή των παλαιότερων samples μετρικής, περιλαμβάνεται μια δομή κυκλικού buffer (CBW_M), όμοια με αυτές του TSY. Το READ_OFFSET είναι όσο το παράθυρο του κινούμενου μέσου όρου, $N_{CP} + 1$, με

αποτέλεσμα να αποθηκεύει ισάριθμα πρόσφατα samples μετρικών. Ανά πάσα στιγμή γράφεται στο buffer η τρέχουσα μετρική $\tilde{M}_{f_{d+1}}$ και διαβάζεται η παλαιότερη $\tilde{M}_{f_{d-N_{CP}}}$. Όπως φαίνεται από την έκφραση παραπάνω, στον υπολογισμό της μετρικής μετέχει η τρέχουσα και η παλαιότερη τιμή της \tilde{M}_d . Ωστόσο, πριν από την είσοδο $\tilde{M}_{f_{d-N_{CP}}}$ της έκφρασης, παρεμβάλλεται ένας MUX, εν είδη μεμβράνης, όμοιας με αυτές που περιγράφηκαν στον TSY. Ο σκοπός της είναι να ελέγχει τα samples που θα μετέχουν στον υπολογισμό της μετρικής, αποκλείοντας άχρηστα samples, ώστε να γίνεται σωστή αρχικοποίηση του accumulator.

Γενικά για το κύκλωμα

Η τελική μετρική που υπολογίζεται, \tilde{M}_{d+1} , διαφέρει από αυτή της (2.4) κατά ένα συντελεστή κλιμάκωσης. Η σχέση της με την $M(d)$ είναι :

$$\tilde{M}(d) = M(d) 2^{(2(\log_2 \text{scale}_{R_f} - \log_2 \text{scale}_P))} = M(d) 2^{(\log_2 \text{scale}_{M_A})}$$

Με την κατάλληλη επιλογή του συντελεστή κλιμάκωσης αποφεύγεται η χρήση fixed point αριθμητικής, καθώς η \tilde{M}_{d+1} αποθηκεύεται σε integer. Η συνολική πολυπλοκότητα των πράξεων του DCR (\tilde{M}_{Ad+1}) $\forall d$ είναι:

$$15xADD, 11xMULT, 2xDIV, 3xSHIFT, 3xLOAD$$

Το συνολικό latency του DCR είναι 0, καθώς υπάρχει απευθείας δρόμος από την είσοδο στην έξοδο (direct I/O path).

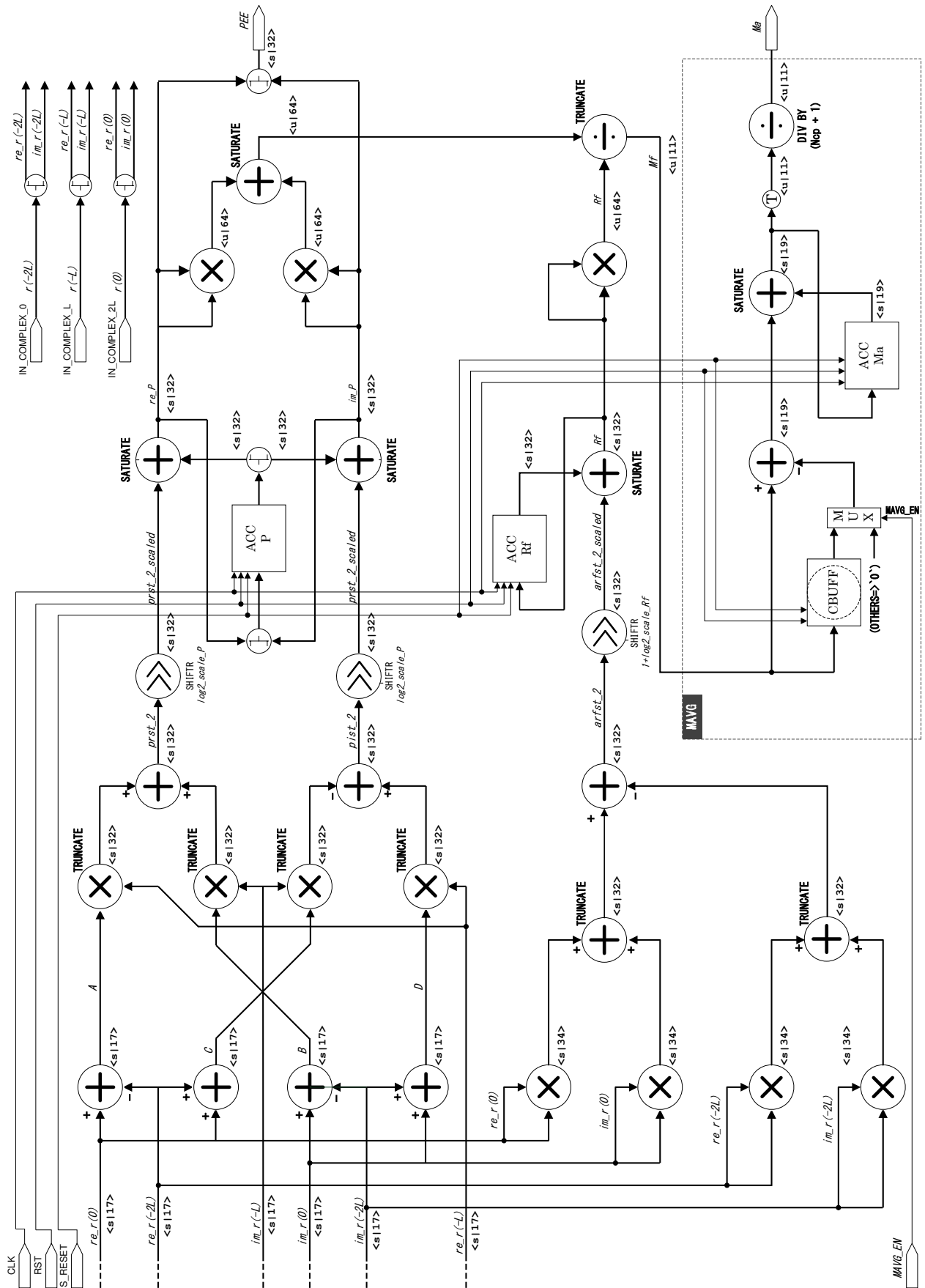
Κατά τη διάρκεια των υπολογισμών πρέπει να ληφθεί ειδική μέριμνα για την αρχικοποίηση των accumulator. Αυτοί αποτελούν νευραλγικό στοιχείο της αριθμητικής του κυκλώματος και πρέπει να διατηρούνται σε συνεπή κατάσταση. Ακόμα και ένα άχρηστο sample να εισέλθει στον υπολογισμό, προκαλεί μία αριθμητική διαταραχή από την οποία ο accumulator μπορεί να μην ανακάμψει ποτέ. Αυτό οδηγεί σε συσσώρευση σφάλματος. Για την παροχή μεγαλύτερης ανεκτικότητας, οι μετρικές μηδενίζονται κάθε φορά που ο TSY εισέρχεται σε sync mode ώστε να εξαλειφθεί οποιαδήποτε αρνητική επίδραση από το παρελθόν.

Για αυτόν τον λόγο υπάρχει η είσοδος σύγχρονου μηδενισμού, `s_reset`, όλων των accumulator που περιλαμβάνονται στον DCR. Ο CO είναι υπεύθυνος για το χρονοισμό τόσο του `s_reset` όσο και των μεμβρανών που υπάρχουν στις εξόδους μνήμης που τροφοδοτούνται σε κάποιο αναδρομικό τύπο. Το `s_reset` μηδενίζει την εσωτερική κατάσταση του DCR, ενώ οι μεμβράνες φροντίζουν για την ομαλή αρχικοποίηση των accumulator, εμποδίζοντας άχρηστα samples που εξέρχονται από τους κυκλικούς buffer, όσο αυτοί αρχικοποιούνται, να εισέλθουν στους υπολογιστές μετρικών.

Οι ακρίβειες όλων των ενδιάμεσων σημάτων φαίνονται στο διάγραμμα του DCR στο σχήμα 4.6.

4.6.4 Decision Unit [TSY.DCU]

```
component decision_unit is
generic(
  METRIC_WIDTH : natural := 0;
  PD_WIDTH : natural := 0;
  METRIC_THRES : natural ; --range 0 to 2**METRIC_WIDTH-1
  OBSERV_WINDOW : natural --range 0 ~ ??? --delay until max decision is made
);
port(
  clk : in std_logic;
  rst : in std_logic;
  m1 : in std_logic_vector(METRIC_WIDTH-1 downto 0); --2*LOG2_SCALE_SQRT
  ena : in std_logic;
  pd : in std_logic_vector(PD_WIDTH-1 downto 0); --2*ACC_WIDTH-1
  ld_valid : in std_logic;
```



Σχήμα 4.6: TSY.DCR datapath και ακριβείες εργασίας

```

    sof_found : out std_logic;
    pd_sof : out std_logic_vector(PD_WIDTH-1 downto 0)
);
end component;

```

Το Decision Unit υλοποιεί τον αλγόριθμο αναζήτησης του μεγίστου σημείου της μετρικής, που περιγράφηκε στην §2.2.3. Η μετρική $\tilde{M}_A(d)$ μεγιστοποιείται σε ένα ολισθαίνων παράθυρο παρατήρησης, εύρους M_{span} samples κάνοντας χρήση ενός κατωφλιού εγκυρότητας της μετρικής $M_{threshold}$. Ο αλγόριθμος που υλοποιείται είναι κατάλληλος για real-time συστήματα όπως το παρόν: Η αναζήτηση του μεγίστου είναι συνεχής μέχρι αυτό να βρεθεί, καθώς συνεχής είναι και η ροή των δεδομένων εισόδου. Οι παράμετροι M_{span} και $M_{threshold}$ ορίζονται ως τηλεπικοινωνιακές παράμετροι και μπορούν να αλλάξουν εύκολα σε ένα άλλο instance του κυκλώματος μέσω του μηχανισμού *Parameter Dispatching*.⁹

Στον DCU εισέρχονται συγχρονισμένα samples των μετρικών $\tilde{M}_A(d)$ και $\tilde{P}(d)$, όπως αυτά εξέρχονται από τον DCR. Όταν βρεθεί το μέγιστο σηκώνεται το σήμα `sof_found` και ταυτόχρονα στην έξοδο `pd_sof` εξέρχεται η τιμή της $\tilde{P}(d)$ που αντιστοιχεί στο μέγιστο. Το `sof_found` ουσιαστικά αναφέρεται στη μετρική που είχε εισέλθει στον DCU πριν από M_{span} samples. Δηλαδή, αν τη στιγμή που συμβαίνει το `sof_found` εισέρχεται η μετρική $\tilde{M}_A(d)$ το μέγιστο αναφέρεται στην $\tilde{M}_A(d - M_{span})$. Αυτή την πληροφορία την «γνωρίζει» ο TSY.CO ο οποίος φροντίζει για την αλλαγή του κυκλώματος σε data mode όταν βρεθεί το μέγιστο. Επιπλέον την «γνωρίζει» και ο CBW_D καθώς πρέπει στην έξοδο του να εμφανιστούν τα samples $r(m)$ που αντιστοιχούν στην μετρική $M_A(d)$ που μεγιστοποιήθηκε.

Περιγραφή λειτουργίας

Ο DCU αποτελείται από ένα ακολουθιακό κύκλωμα που λειτουργεί όπως ένας βρόχος εντολών σε μία τυπική γλώσσα προγραμματισμού, που εκτελείται πάνω στα samples εισόδου `m1_u`. Υπάρχουν οι εξής μεταβλητές που αλληλεπιδρούν με το βρόχο:

`max_found` → σηματοδοτεί το γεγονός ότι έχει βρεθεί υποψήφιο μέγιστο,
`max_val` → η τιμή του υποψήφιου μεγίστου,
`max_since` → μετράει τα samples για τα οποία δεν έχει βρεθεί νέο μέγιστο,
`max_p_val` → η τιμή της \tilde{P} μετρικής που αντιστοιχεί στο υποψήφιο μέγιστο.

Για την εύρεση του μεγίστου εκτελούνται οι εξής διαδικασίες.

Γίνεται συνεχής έλεγχος αν το `m1_u` είναι μεγαλύτερο από το κατώφλι. Όταν συμβεί αυτό, το `max_found` γίνεται '1' και καταχωρούνται στα `max_val` και `max_p_val` οι τρέχουσες τιμές των $\tilde{M}_A(d)$ και $\tilde{P}(d)$ αντίστοιχα. Η `max_since` αρχίζει να αυξάνεται σε κάθε επόμενο sample αν αυτό δεν ξεπεράσει το `max_val`. Αντιθέτως αν υπάρξει κάποιο sample που είναι μεγαλύτερο του `max_val`, μηδενίζεται το `max_since` και ανανεώνονται τα `max_val` και `max_p_val`. Ταυτόχρονα με την παραπάνω διαδικασία γίνεται συνεχής έλεγχος εάν το `max_since` είναι μεγαλύτερο ή ίσο από το M_{span} . Όταν συμβεί αυτό σηκώνεται το `sof_found` για να σηματοδοτηθεί η εύρεση του μεγίστου. Επιπλέον μηδενίζονται τα `max_val`, `max_since`, `max_found` ενώ το `max_p_val` περνάει στην έξοδο `pd_sof`.

Ο CO έχει τη δυνατότητα να επαναφέρει (reset) ή να απενεργοποιήσει τη λειτουργία της αναζήτησης του DCU, με τα σήματα `ld_valid` και `ena`. Όταν κατεβαίνει το σήμα `ld_valid`, δεν λαμβάνονται υπόψη τα samples που εισέρχονται και ο DCU αρχικοποιείται (reset) μηδενίζοντας τις τιμές των `max_val`, `max_since`, `max_found`. Δηλαδή το module προετοιμάζεται για την επόμενη αναζήτηση. Το `ld_valid` κατεβαίνει κατά τη λειτουργία του κυκλώματος, όταν ο TSY είναι σε data mode.

⁹ Στην πραγματικότητα, (βλ. §2.2.3) η τηλεπικοινωνιακή παράμετρος είναι η M_{rthres} που ορίζεται ως ποσοστό επί της μέγιστης τιμής της μετρικής, δηλαδή $2^{(\log_2\text{-scale}_{M_A})} = 1024$. Μέσω του μηχανισμού *Asset Dispatching* που εξηγείται στην §5.1.3 γίνεται η μετατροπή από M_{rthres} σε $M_{threshold}$.

Όταν κατεβαίνει το `ena`, ο DCU παγώνει στην προηγούμενη κατάσταση και δεν λαμβάνονται υπόψη τα `samples` εισόδου. Αυτό συμβαίνει όταν παγώνει ολόκληρος ο TSY.¹⁰

4.6.5 Controller [TSY.CO]

```

component sync_controller is
generic (
  -- symbol size in samples
  SZ_SYM : natural ;
  -- usefull symbol size
  SZ_USYM : natural ;
  -- frame size in symbols
  SZ_FRAME : natural ;
  -- number of frames we search for the peak (Z)
  SEARCH_SPAN : natural ;
  -- linear pointer width
  PTR_WIDTH : natural;
  -- sample frame counter width (must be >= log2(SZ_SYM*SZ_FRAME)
  SA_FR_CNT_WIDTH: natural
);
port (
  clk: in std_logic;
  rst: in std_logic;

  in_valid: in std_logic;
  in_sof_found: in std_logic;

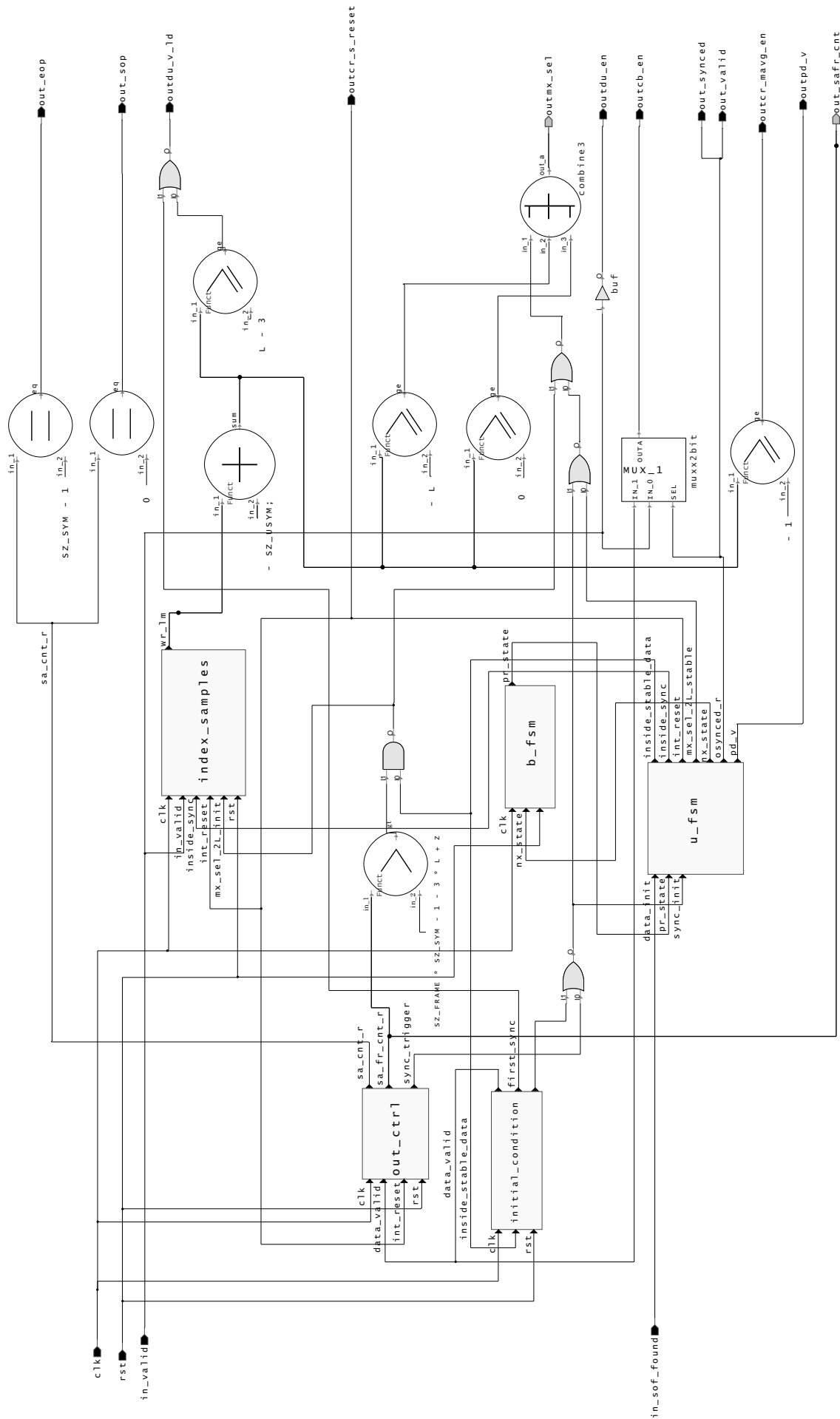
  -- cbuffer control
  outcb_en: out std_logic;
  -- arc_comp control
  outpd_v : out std_logic; --valid pd for arg computer
  -- data mux
  outmx_sel: out std_logic_vector(2 downto 0); -- [0 X X] | [X L X] | [X X 2L] || (0,L,2L)->(0,1)
  -- cruncher
  outcr_mavg_en : out std_logic;
  outcr_s_reset : out std_logic;
  -- decision
  outdu_en: out std_logic;
  -- outdu_ld: out std_logic_vector( PTR_WIDTH-1 downto 0); --unsigned
  outdu_v_ld : out std_logic;
  --external
  out_synced: out std_logic;
  out_valid: out std_logic;
  out_sop: out std_logic;
  out_eop: out std_logic;
  out_safr_cnt: out std_logic_vector(SA_FR_CNT_WIDTH-1 downto 0) -- unsigned
);
end component;

```

Ο CO είναι ο γενικός συντονιστής των λειτουργιών του TSY. Είναι υπεύθυνος για το χρονισμό των DCR, DCU, ARG, φροντίζοντας για την ομαλή μετάβαση μεταξύ των δύο κύριων καταστάσεων λειτουργίας του κυκλώματος, `data` και `sync`. Κατά μία έννοια, ο CO «γνωρίζει» λεπτομέρειες τις αρχιτεκτονικής του συστήματος, όπως τα μεγέθη των CBW, την καθυστέρηση του `sof_found` συμβάντος, τους χρόνους αρχικοποίησης του DCR. Μεταξύ άλλων, χειρίζεται τον μηχανισμό των μεμβρανών τόσο στην είσοδο του DCR όσο και στην είσοδο του DCR.MVG. Χειρίζεται, επίσης, τα σήματα του Avalon Interface στην έξοδο του TSY που ελέγχουν τη ροή και την εγκυρότητα των `sample` εξόδου.

Βασικές οντότητες που απαρτίζουν τον CO είναι η μηχανή καταστάσεων (διεργασίες `u_fsm`, `b_fsm`), οι συνδεδεμένοι μετρητές `sample sa_cnt_r`, `sa_fr_cnt_r` που μετράνε κατά τη διάρκεια του `data mode`

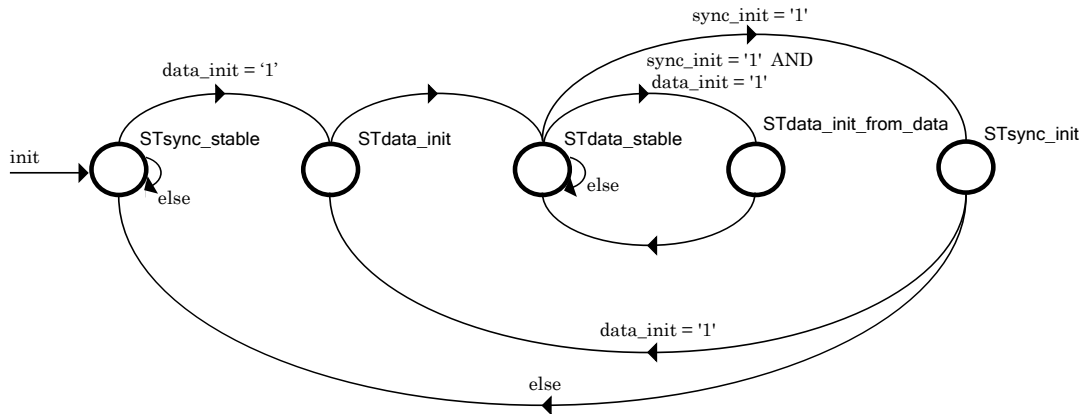
¹⁰Αυτό μπορεί να συμβεί π.χ. στη λειτουργία TDD.



Σχήμα 4.7: TSY.CO Top-Level διάγραμμα

(διεργασία `out_ctrl`), ένας βοηθητικός μετρητής `samples (wr_lm)` που μετράει κατά τη διάρκεια του `sync mode` (διεργασία `index_samples`) και συνδυαστική λογική που παράγει σήματα ελέγχου από τις τιμές των `wr_lm`, `sa_cnt`, `sa_fr_cnt`. Το top-level διάγραμμα για τον CO απεικονίζεται στο Σχήμα 4.7.

Μηχανή Καταστάσεων



Σχήμα 4.8: TSY.CO διάγραμμα καταστάσεων

Η μηχανή καταστάσεων είναι τύπου Mealy,¹¹ με τις εξής 5 καταστάσεις

`STsync_init`, `STsync_stable`, `STdata_init`, `STdata_stable`, `STdata_init_from_data`

Στο Σχήμα 4.8 φαίνεται το διάγραμμα καταστάσεων της μηχανής.

Οι δύο κύριες καταστάσεις είναι `STsync_stable` και `STdata_stable` για το `sync` και `data mode` του κύκλωματος αντίστοιχα. Οι υπόλοιπες καταστάσεις είναι μεταβατικές, δηλαδή η μηχανή δεν κάθεται περισσότερο από ένα κύκλο σε αυτές.

Η μηχανή ξεκινά `STsync_stable` και αναμένεται η εύρεση του SOF. Όταν αυτό συμβεί, γίνεται μετάβαση σε `STdata_init` και κατόπιν σε `STdata_stable`. Όταν τελειώσει το frame η μηχανή μεταβαίνει σε `STsync_init` εκτός και αν η αρχή του επόμενου frame έχει βρεθεί αυτή τη στιγμή, οπότε μεταβαίνει σε `STdata_init_from_data` και έπειτα σε `STdata_stable`. Μετά την κατάσταση `STsync_init` μεταβαίνει σε `STsync_stable`, εκτός και αν έχει βρεθεί η αρχή του frame αυτή τη στιγμή, οπότε μεταβαίνει σε `STdata_init` και από εκεί σε `STdata_stable`. Με αυτόν τον τρόπο έχουν προβλεφθεί όλες οι δυνατές μεταβάσεις σε `data mode`, ικανοποιώντας όλες τις ακραίες περιπτώσεις έναρξης του frame.

Υπενθυμίζεται ότι η αρχή του frame μπορεί να βρίσκεται μετά από τυχαίο χρονικό διάστημα καθυστέρησης σε σχέση με τέλος του προηγούμενου. Οι δύο ειδικές περιπτώσεις είναι όταν η καθυστέρηση είναι 0 ή 1 `sample` μετά το πέρας του frame. Παρόλο που όπως αναφέρθηκε στο κεφ 2 υπάρχει περιθώριο σφάλματος όσων αφορά το SOF, είναι προτιμότερο να μην εισάγεται από το ίδιο το σύστημα.

Όταν το κύκλωμα βρίσκεται σε `sync mode`, οι μετρικές υπολογίζονται από τον DCR και ο DCU αναζητά το μέγιστο. Μόλις αυτό συμβεί σηκώνει το σήμα `sof_found`. Αυτό έχει ως αποτέλεσμα την μετάβαση σε κατάσταση `STdata_init` ή `STdata_init_from_data`. Η δεύτερη μετάβαση προκαλείται όταν το `sof_found` συμβαίνει στο νωρίτερο δυνατό σημείο, 2 `samples` πριν το τέλος του frame, δηλαδή από το σημείο που σηκώνεται το τελευταίο `out_eop` του frame. Σε οποιαδήποτε από αυτές τις καταστάσεις σηκώνεται το σήμα `outpd_v` καθώς στην έξοδο του ARG έχει υπολογιστεί η εκτίμηση της διαφοράς φάσης.¹²

¹¹Δηλαδή η έξοδος της μηχανής κατάστασης εξαρτάται τόσο από την τρέχουσα κατάσταση, όσο και από την τρέχουσα είσοδο.

¹²Αυτή η μετάβαση αντιστοιχεί στην περίπτωση που το επόμενο frame έρχεται ακριβώς μετά το τρέχων.

Για να είναι εφικτή η εύρεση του επόμενου frame όταν η καθυστέρηση είναι η μικρότερη δυνατή, ο υπολογισμός των μετρικών και η αναζήτηση του μεγίστου αρχίζουν να εκτελούνται παράλληλα με την έξοδο των τελευταίων samples του frame, λίγο πριν την μετάβαση σε sync mode.¹³ Επίσης αυτή τη στιγμή μηδενίζονται οι συνδεδεμένοι μετρητές και οι accumulator του DCR. Η εσωτερική μεμβράνη του DCR.MVG καθώς και οι εξωτερικές μεμβράνες του DCR κλείνουν κρατώντας την εσωτερική κατάσταση του DCR μηδενική μέχρι την επόμενη διαδικασία αρχικοποίησης.

Συνδεδεμένοι Μετρητές

Οι συνδεδεμένοι μετρητές είναι παρόμοιοι με αυτούς που περιγράφηκαν στον FRG. Εσωτερικά, στο process `out_ctrl`, υπάρχει :

- ο `sa_cnt`, που μετράει samples ανά σύμβολο με εύρος $[0, N_s - 1]$,
- ο `sy_cnt`, που μετράει OFDM σύμβολα με εύρος $[0, R - 1]$,
- και ο `sa_fr_cnt`, που μετράει τα συνολικά samples ενός OFDM συμβόλου με εύρος $[0, R * N_s - 1]$.¹⁴

Στην έξοδο του process εκχωρούνται στα σήματα `sa_cnt_r` και `sa_fr_cnt_r` οι αντίστοιχοι μετρητές.

Οι μετρητές αυτοί χρησιμοποιούνται κατά το data mode και αρχικοποιούνται με τις καταστάσεις `STdata_init`, `STdata_init_from_data`. Ένα συνδυαστικό κύκλωμα παράγει τα σήματα εξόδου `out_sop`, `out_eop`, από τις τιμές του `sa_cnt_r`. Ο `sa_fr_cnt_r` περνάει στην έξοδο του TSY ώστε να αριθμοδοτεί χρονικά τα samples που περνάνε στον FCO. Επίσης από το process των μετρητών σηκώνεται το σήμα `sync_trigger` που σηματοδοτεί μετάβαση σε `STsync_init`, όταν τελειώνει το frame. Λίγο νωρίτερα σηκώνεται ένα άλλο σήμα (`mx_sel_2L_init`), με τη βοήθεια του `sa_fr_cnt`, το οποίο φροντίζει για δύο πολύ σημαντικές λειτουργίες:

- ανοίγει την μεμβράνη της εισόδου $r(-2L)$ του DCR στο κατάλληλο χρονικό σημείο και
- ξεπαγώνει τον μετρητή `wr_lm`

Το αποτέλεσμα είναι η έναρξη της διαδικασίας αρχικοποίησης του DCR, **παράλληλα** με την εξαγωγή των τελευταίων samples του frame.

Ο Μετρητής `wr_lm`

Ο μετρητής `wr_lm`, χρησιμοποιείται για το χρονισμό των διαδικασιών κατά την έναρξη του sync mode. Διάφορα συνδυαστικά κυκλώματα φροντίζουν για τον έλεγχο σημάτων σύμφωνα με τις τιμές του `wr_lm`. Όταν το κύκλωμα βρίσκεται σε data mode, ο `wr_lm` είναι παγωμένος στο μηδέν και ξεπαγώνει λίγο πριν την έναρξη του sync mode. Αρχικά ανοίγει τη μεμβράνη της εισόδου $r(-L)$ του DCR, αφού έχουν περάσει L samples από τότε που άνοιξε η $r(-2L)$. Στη συνέχεια σηκώνει το σήμα `out_cr_avg_en` το οποίο τροφοδοτείται στον DCR.MVG και ανοίγει την εσωτερική μεμβράνη του CBW που βρίσκεται εκεί καθώς στον επόμενο κύκλο θα εξέλθει το πρώτο έγκυρο δεδομένο από αυτόν. Στον επόμενο κύκλο ανοίγει και την τρίτη μεμβράνη της εισόδου $r(0)$ του DCR. Στον επόμενο κύκλο θα βγει η πρώτη έγκυρη μετρική \tilde{M}_A , από τον DCR. Κατόπιν σηκώνει το σήμα `du_v_ld` το οποίο ενεργοποιεί τον DCU. Το σημείο ενεργοποίησης είναι $M_{span} - 3$ samples πριν το τέλος του frame, ώστε το πρώτο πιθανό `sof_found` συμβάν, να έρθει ακριβώς 2 sample νωρίτερα από το τέλος του frame. Υπενθυμίζεται ότι από τη στιγμή που συμβαίνει το `sof_found`, το πρώτο έγκυρο δεδομένο βγαίνει στην έξοδο του TSY μετά από δύο κύκλους.

¹³Την εκτίμηση αυτή θα την αποθήκευσει ο FCO, που βρίσκεται μετά τον TSY για τον συγχρονισμό συχνότητας.

¹⁴Ο `sa_fr_cnt` θα μπορούσε να παραχθεί και με έναν πολλαπλασιαστή από τους άλλους δύο μετρητές, αλλά προτιμήθηκε ένας τρίτος μετρητής μιας και στο FPGA δεν υπάρχουν πολλοί πολλαπλασιαστές, ενώ αντίθετα υπάρχουν αρκετοί καταχωρητές και adders.

Ο εξτρά κύκλος προέρχεται από την καθυστέρηση ανταπόκρισης του DCU στο `du_v_1d`. Με αυτόν τον τρόπο παρέχεται προστασία από το να βρεθεί το μέγιστο νωρίτερα από το όριο που αναφέρθηκε. Αυτό θα μπορούσε να συμβεί σε κανάλια με θόρυβο και θα οδηγούσε στην απώλεια του frame, καθώς ο DCU κρατάει το `sof_found` πάνω μόνο για ένα κύκλο.

4.6.6 ARG Computer [TSY.ARG]

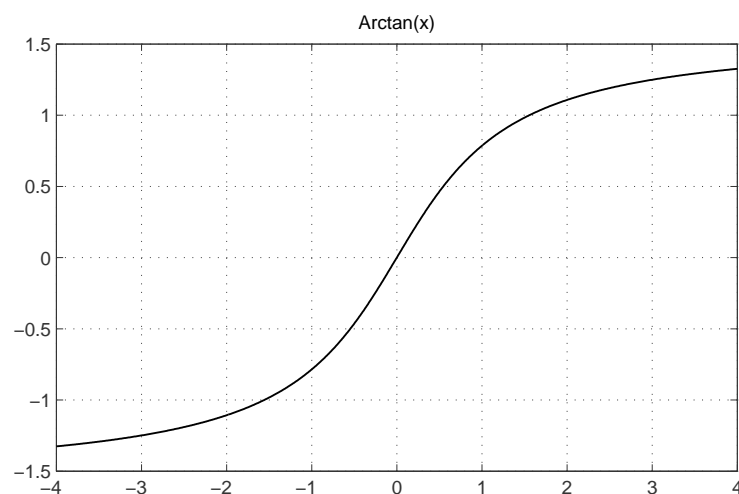
```
component arg_computer is
  generic(
    CMPLX_WIDTH : natural --32
  );
  port(
    complex_in : in std_logic_vector(CMPLX_WIDTH-1 downto 0); --complex;
    arg : out std_logic_vector(15 downto 0)
  );
end component;
```

Ο ARG computer είναι ο υπολογιστής του πρωτεύοντος ορίσματος ($Arg(x)$) ενός μιγαδικού αριθμού. Στην προκειμένη, ο αριθμός είναι ο $\tilde{P}(d)$ και το πρωτεύον όρισμά του είναι η εκτίμηση της διαφοράς φάσης μεταξύ πρώτου και δεύτερου μισού του training symbol, που αποτελεί την coarse εκτίμηση συχνότητας.

Το κύκλωμα αποτελείται μόνο από συνδυαστική λογική, επιτυγχάνοντας ταχύ υπολογισμό του Arg , έχοντας I/O latency 0. Κάθε επιπλέον στάδιο latency σε αυτό το κρίσιμο στάδιο του συγχρονισμού προσθέτει I/O latency σε ολόκληρο το RX datapath διότι αυξάνεται το μέγεθος του κυκλικού buffer CBW_D, μέσα από τον οποίο ρέουν τα samples εισόδου.

Καθώς το Arg ενός μιγαδικού αριθμού μπορεί να υπολογιστεί με τη βοήθεια της $arctan$, το παρόν module υπολογίζει τη συνάρτηση αυτή. Υλοποιείται ένας αλγόριθμος πολυωνυμικής προσέγγισης όπως περιγράφεται στο [Raj+06]. Η ακρίβεια που επιτυγχάνεται είναι ικανοποιητική για την τρέχουσα εφαρμογή, ενώ η έκταση του κυκλώματος είναι σχετικά μικρή.

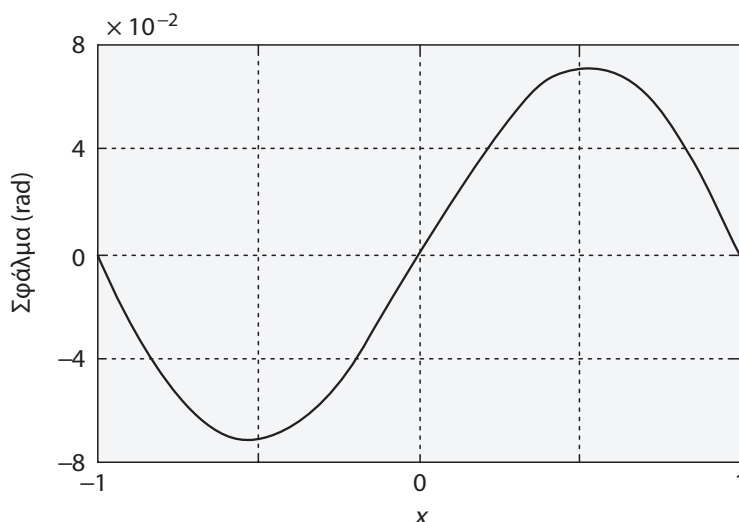
Κατά το στάδιο του σχεδιασμού, εξετάστηκε ο κλασικός αλγόριθμος CORDIC, που υπολογίζει την $arctan$ μόνο με προσθέσεις και ολισθήσεις. Ωστόσο απορρίφθηκε γρήγορα καθώς η έκταση που θα καταλάμβανε το κύκλωμα και το latency που θα εισήγαγε, θα επιβάρυναν αρκετά το κύκλωμα.



Σχήμα 4.9: Γράφημα της $arctan$ στο διάστημα $[-4,4]$

Το πρωτεύον όρισμα ενός μιγαδικού αριθμού $Z = I + jQ$ ορίζεται ως $\angle(Z) = \text{Arg}(Z) = \arctan(Q/I)$ με $\text{Arg}(Z) \in (-\pi, \pi]$.¹⁵ Το πεδίο ορισμού της $y = \arctan(x)$ είναι $x \in (-\infty, +\infty)$ και το πεδίο τιμών της $y \in (-\pi, +\pi]$. Το γράφημα της \arctan (στο διάστημα $[-4,4]$) παρουσιάζεται στο Σχήμα 4.9. Η $\arctan(Q/I)$ μπορεί να θεωρηθεί σαν συνάρτηση δύο μεταβλητών με πεδίο ορισμού όλο το μιγαδικό επίπεδο. Στην παρούσα προσέγγιση, η \arctan υπολογίζεται για γωνίες στο $[-\pi/4, \pi/4]$ και στη συνέχεια γίνεται χρήση τριγωνομετρικών ταυτοτήτων για την ανακατασκευή της, με γραμμικό τρόπο, σε όλο το πεδίο τιμών.

Προσέγγιση της \arctan



Σχήμα 4.10: Σφάλμα προσέγγισης α' τάξης (E_1) της \arctan

Μία από τις κλάσεις μεθόδων που υπάρχουν για την προσέγγιση της \arctan , είναι η προσέγγιση με πολυώνυμα δευτέρου και τρίτου βαθμού βασισμένα στην παρεμβολή Lagrange και στο κριτήριο minimax για τον υπολογισμό των συντελεστών. Όπως εξηγείται στο [Raj+06] κάνοντας εφαρμογή της παρεμβολής Lagrange, μπορεί να υπολογιστεί μια προσέγγιση πολυωνύμου πρώτου βαθμού, στο διάστημα $[-1, 1]$. Αποδεικνύεται ότι:

$$A_1(x) = \frac{\pi}{4} x \quad x \in [-1, 1]$$

Το σφάλμα προσέγγισης σε αυτή την περίπτωση (Σχ. 4.10) είναι $E_1(x) = \arctan(x) - A_1(x)$ και παρουσιάζει μέγιστο στο $x_{max} = \pm(4/\pi - 1)^{1/2}$ με $E_1(x_{max}) \simeq 0.07 \text{ rad}$ (4°). Αυτή η αρχική γραμμική προσέγγιση, δεν παρέχει την επιθυμητή ακρίβεια.

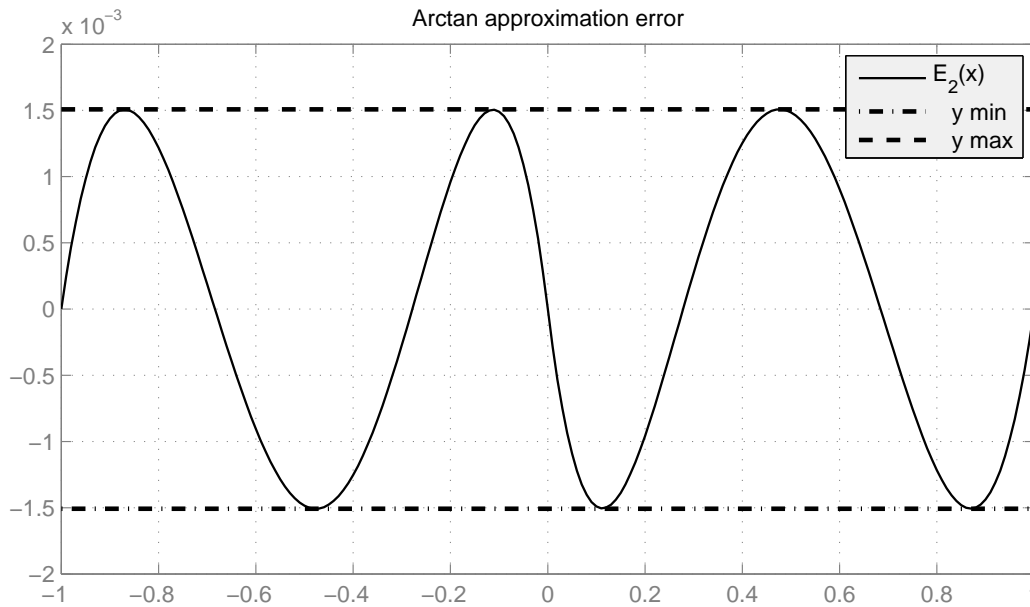
Η μέθοδος που προτείνεται στο [Raj+06] για την επίτευξη καλύτερης προσέγγισης βασίζεται στην παρατήρηση ότι η συνάρτηση σφάλματος, $E_1(x)$, είναι τετραγωνικής φύσεως, συνεπώς επιχειρείται εκ νέου μία προσέγγιση αυτής. Παρουσιάζονται διάφοροι τύποι τόσο πολυωνυμικών, όσο και κλασματικών προσεγγίσεων, με τον καθένα να επιτυγχάνει διαφορετική ακρίβεια με διαφορετικό υπολογιστικό κόστος.

Μετά από εξέταση των προτεινόμενων μεθόδων, οι κλασματικές απορρίφθηκαν για την παρούσα υλοποίηση, μιας και η διαίρεση είναι μία πράξη που πρέπει να αποφεύγεται λόγω της υπολογιστικής της πολυπλοκότητας. Επιλέχθηκε μία πολυωνυμική προσέγγιση 3^{ου} βαθμού, που επιτυγχάνει μέγιστο σφάλμα $\max(E_2(x)) = 0.0015 \text{ rad}$ (0.086°).

Συγκεκριμένα, υποτίθεται ότι η μορφή της $E_1(x)$ είναι

$$\psi(x) = x(x-1)(ax-b) \quad x \in [0, 1]$$

¹⁵Όπως είναι γνωστό ο μιγαδικός αριθμός Z μπορεί να γραφεί $Z = |Z| \exp(j \text{Arg}(Z))$.



Σχήμα 4.11: Σφάλμα προσέγγισης γ' τάξης (E_2) της \arctan

Κάνοντας χρήση του κριτηρίου minimax αποκτάται το εξής πολυώνυμο:

$$\arctan(x) \simeq A_2(x) = \frac{\pi}{4} x - x(|x| - 1)(0.0663|x| + 0.2447) \quad x \in [-1, 1]$$

Το σφάλμα της προσέγγισης $E_2(x) = \arctan(x) - A_2(x)$ απεικονίζεται στο Σχήμα 4.11.

Κριτήριο Minimax

Για τον υπολογισμό των συντελεστών του πολυωνύμου χρησιμοποιείται το κριτήριο minimax:

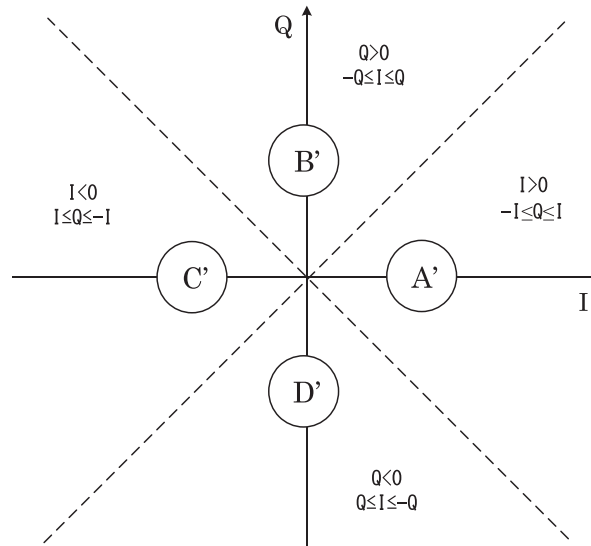
$$J = \min_{a,b} \left\{ \max_{0 \leq x \leq 1} \{|E_1(x) - (x(x-1)(ax-b))|\} \right\}$$

Συγκεκριμένα, το μέγιστο απόλυτο σφάλμα μεταξύ της $E_1(x)$ και της πολυωνυμικής προσέγγισης τρίτου βαθμού με την υποτιθέμενη μορφή, εκφράζεται ως συνάρτηση των παραμέτρων a, b . Στη συνέχεια το σφάλμα αυτό ελαχιστοποιείται ως προς a και b και από εκεί προκύπτουν οι βέλτιστες τιμές των παραμέτρων, που είναι και μοναδικές. Αυτή η διαδικασία είναι υπολογιστικά πολύπλοκη και συνήθως επιτελείται με τη βοήθεια προγραμμάτων συμβολικής ανάλυσης.

Ανακατασκευή στον πλήρη κύκλο

Η προσέγγιση $A_2(x)$ ισχύει για γωνίες στο διάστημα $y \in [-\pi/4, \pi/4]$ που έχουν εφαπτομένη $x = \tan(y) \in [-1, 1]$, δηλαδή στο Α' τεταρτημόριο, σύμφωνα με την αρίθμηση του Σχήματος 4.12. Για τον υπολογισμό της \arctan στα υπόλοιπα τεταρτημόρια έγινε χρήση των τριγωνομετρικών της ιδιοτήτων, με αποτέλεσμα την αναγωγή του υπολογισμού τους στο Α' τεταρτημόριο. Οι αναγωγικοί τύποι είναι:

$$\begin{aligned} Z_A &= A_2\left(\frac{Q}{I}\right) \\ Z_B &= \frac{\pi}{2} - A_2\left(\frac{I}{Q}\right) \\ Z_C &= s(Q)\pi + A_2\left(\frac{Q}{I}\right) \\ Z_D &= -\frac{\pi}{2} - A_2\left(\frac{I}{Q}\right) \end{aligned}$$

Σχήμα 4.12: Αρίθμηση τεταρτημορίων για την ανακατασκευή της \arctan

όπου η συνάρτηση προσήμου ενός πραγματικού αριθμού είναι

$$s(x) = \begin{cases} +1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

Στον πίνακα 4.8 απεικονίζονται οι ανεπτυγμένοι τύποι προσέγγισης για τα τέσσερα τεταρτημόρια, έχοντας αντικαταστήσει τις τιμές των σταθερών.

Τετ/ριο	Τύπος
A'	$Z_A = \frac{Q}{I} \left\{ 1.0301 - s(Q) \frac{Q}{I} \left[0.1784 + 0.0663 s(Q) \frac{Q}{I} \right] \right\}$
B'	$Z_B = 1.570 - \frac{I}{Q} \left\{ 1.0301 - s(I) \frac{I}{Q} \left[0.1784 + 0.0663 s(I) \frac{I}{Q} \right] \right\}$
C'	$Z_C = 3.1416 s(Q) + \frac{Q}{I} \left\{ 1.0301 + s(Q) \frac{Q}{I} \left[0.1784 - 0.0663 s(Q) \frac{Q}{I} \right] \right\}$
D'	$Z_D = -1.570 - \frac{I}{Q} \left\{ 1.0301 + s(I) \frac{I}{Q} \left[0.1784 - 0.0663 s(I) \frac{I}{Q} \right] \right\}$

Πίνακας 4.8: Προσέγγιση της \arctan σε όλα τα τεταρτημόρια

Υλοποίηση

Για την υλοποίηση των ανωτέρω τύπων η μορφή τους εξελίχθηκε λίγο ακόμα, ώστε να γίνει δυνατή η εκμετάλλευση της ομοιομορφίας που είναι προφανές ότι υπάρχει, με σκοπό την εξοικονόμηση υλικού.

Αρχικά γίνεται εύρεση του τεταρτημορίου, από ένα κύκλωμα συγκριτών μεταξύ I και Q και το αποτέλεσμα εξέρχεται στο σήμα Q . Στον πίνακα 4.9 αναγράφονται οι σχέσεις εύρεσης του τεταρτημορίου και οι τιμές του σήματος Q . Στη συνέχεια, υπολογίζεται η ποσότητα

$$Z_X = K \{ 1.0501 - L K [0.1784 + 0.0663 L K] \}$$

όπου K, L μεταβλητές εξαρτώμενες από το τεταρτημόριο. Οι τιμές των K, L αναγράφονται στον πίνακα 4.10. Καθώς η μεταβλητή K είναι το κλάσμα Q/I ή I/Q , ανάλογα με την περίπτωση, χρησιμοποιήθηκε

ένας διαιρέτης με κατάλληλους πολυπλέκτες στην είσοδό του. Με αυτόν τον τρόπο αποφεύχθηκε η εισαγωγή δεύτερου διαιρέτη στο κύκλωμα. Η μεταβλητή L υπολογίζεται από ένα MUX συναρτήσει του τεταρτημορίου και του εκάστοτε προσήμου που πρέπει να ληφθεί.

Τελικά γίνεται ανακατασκευή της \arctan σε όλα τα τεταρτημόρια και υπολογίζεται η έξοδος Z_O συναρτήσει του Z_X . Οι σχέσεις ανακατασκευής απεικονίζονται στον πίνακα 4.10. Στην έξοδο υπάρχει επίσης ένας MUX που επιλέγει το κατάλληλο τμήμα της Z_O ανάλογα με το Q .

Στο Σχήμα 4.13 απεικονίζεται η αρχιτεκτονική του κυκλώματος, καθώς και οι ακρίβειες εργασίας των ενδιάμεσων σημάτων.

Τετ/μόριο	Σχέση		Q
A'	$I > 0$	$-I \leq Q \leq I$	'qA'
B'	$Q > 0$	$-Q < I < Q$	'qB'
C'	$I < 0$	$I \leq Q \leq -I$	'qC'
D'	$Q < 0$	$Q < I < -Q$	'qD'

Πίνακας 4.9: Σχέσεις εύρεσης τεταρτημορίου

Τετ/μόριο	K	L	Z_O
A'	$\frac{Q}{I}$	$s(Q)$	Z_X
B'	$\frac{I}{Q}$	$s(I)$	$1.570 - Z_X$
C'	$\frac{Q}{I}$	$-s(Q)$	$s(Q) 3.1416 + Z_X$
D'	$\frac{I}{Q}$	$-s(I)$	$1.570 - Z_X$

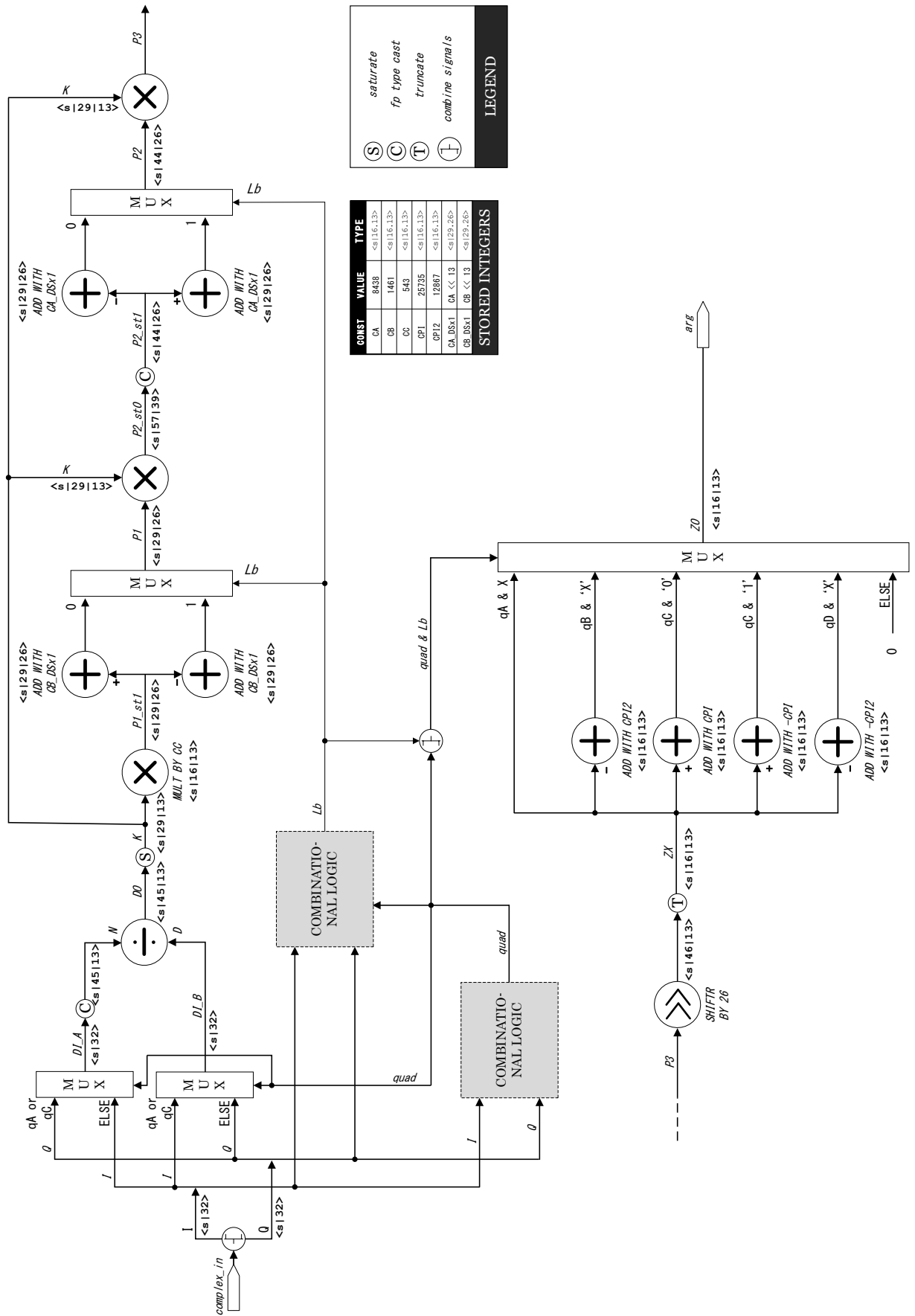
Πίνακας 4.10: Σχέσεις ανακατασκευής \arctan και τιμές των K, L

Αριθμητική - Ακρίβειες

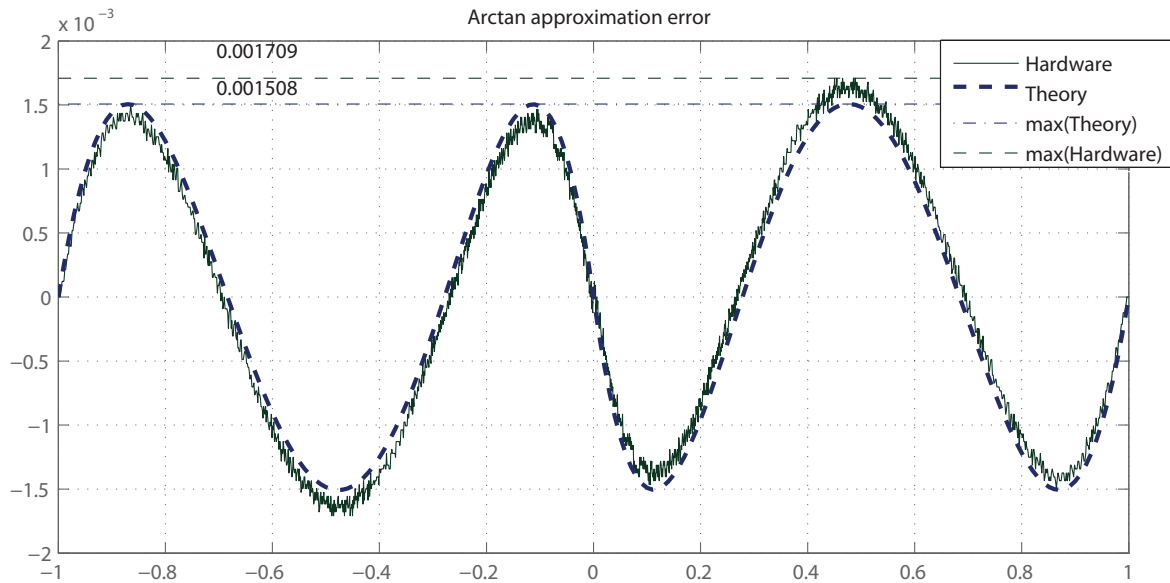
Στην είσοδο του κυκλώματος ο μιγαδικός `complex_in` έχει εύρος 64 bit, όσο και η αντίστοιχη έξοδος του DCR, με το κάθε στέλεχος του αριθμού να είναι σε μορφή `<s | 32>`. Στην έξοδο `arg` εξέρχεται το όρισμα του αριθμού ως `<s | 16. 13>`. Η επιλογή των 13 bit για το κλασματικό μέρος έγινε ώστε να μπορεί να αναπαραστήσει το όρισμα σχεδόν με την ακρίβεια, 0.0015, που έχει επιτευχθεί από τους προσεγγιστικούς τύπους, ενώ τα δύο bit ακέραιου μέρους σε προσημασμένο αριθμό είναι αρκετά για να καλύψουν το δυναμικό εύρος, $[-\pi, \pi]$.

Συγκεκριμένα εάν υποθέσουμε στρογγυλοποίηση `floor` το σφάλμα αναπαράστασης σε `<s | 16. 13>` είναι $2^{-13} = 0.00012$. Παρόλο που αυτό το σφάλμα είναι πολύ μικρό σε σχέση με το σφάλμα προσέγγισης και θα μπορούσαν να είχα επιλεγεί λιγότερα bit, η ακρίβεια εργασίας πρέπει να είναι στα 13 bit για την επίτευξη της επιθυμητής ακρίβειας. Συνεπώς κρατήθηκε αυτή η ακρίβεια ως τύπος αποθήκευσης ώστε να αποφευχθεί η εισαγωγή σφαλμάτων στρογγυλοποίησης κατά την αποθήκευση.

Ακρίβεια εξόδου Η επιθυμητή ακρίβεια εργασίας στο σήμα εξόδου βρέθηκε με δοκιμές. Για τον υπολογισμό των σφαλμάτων χρησιμοποιήθηκε βοηθητικό πρόγραμμα MATLAB. Το σφάλμα προσέγγισης τόσο για το υλοποιημένο, όσο και για το θεωρητικό μοντέλο απεικονίζονται στο Σχήμα 4.14. Το μέγιστο σφάλμα (ακρίβεια εξόδου) στο υλοποιημένο κύκλωμα είναι **0.00171**, λίγο μεγαλύτερο από το θεωρητικό σφάλμα του τύπου προσέγγισης. Ωστόσο αυτή η ακρίβεια θεωρείται ικανοποιητική για τους σκοπούς του κυκλώματος.



Σχήμα 4.13: TSY.ARG datapath και ακρίβειες εργασίας



Σχήμα 4.14: Σφάλμα του ARG computer

4.7 Frequency Compensator [FCO]

Ο FCO εκτελεί την διαδικασία της αντιστάθμισης συχνότητας ή συγχρονισμού στη συχνότητα. Για να γίνει αυτό δέχεται ως είσοδο την διαφορά φάσης, $\hat{\phi}$ μεταξύ δύο samples με απόσταση $N_o/2$, όπως έχει εκτιμηθεί από τον TSY. Στη συνέχεια αντισταθμίζει τα samples εισόδου, $r(n)$, σύμφωνα με τον τύπο:

$$r_c(n) = r(n) \exp(-j \frac{2n\hat{\phi}}{N_o}) \quad (4.1)$$

όπου $r_c(n)$ τα εξερχόμενα αντισταθμισμένα samples, n ο χρονικός δείκτης των samples και N_o το μήκος του ωφέλιμου OFDM συμβόλου. Υπενθυμίζεται, από το Κεφάλαιο 2, ότι για την coarse εκτίμηση ολίσθησης συχνότητας είναι $\hat{D}_f = \frac{\hat{\phi}}{\pi T_o}$. Επιπλέον ισχύει $|\hat{\phi}| < \pi \Rightarrow |\hat{D}_f| < \Delta_f$.¹⁶

Περιγραφή λειτουργίας

Ο FCO, είναι ένα ως επί το πλείστον, αριθμητικό module. Διαθέτει δύο λογικές εισόδους: την `in_phi`, όπου δέχεται την εκτίμηση της διαφοράς φάσης και τις `in_real`, `in_imag`, `in_n`, όπου δέχεται τα χρονικά samples του συγχρονισμένου στο χρόνο OFDM σήματος και το χρονικό δείκτη που τους αντιστοιχεί. Στις εξόδους `out_real`, `out_imag`, παρέχει samples συγχρονισμένα στη συχνότητα. Τόσο στην είσοδο, όσο και στην έξοδο υλοποιείται το Avalon ST Interface. Λόγω των δύο λογικών εισόδων, υπάρχει ξεχωριστό σήμα `valid` για κάθε μία από αυτές.

Η εκτίμηση φάσης παρέχεται σε κάποια χρονική στιγμή για έναν κύκλο στην είσοδο `in_phi` και τα samples που ακολουθούν μέχρι το πέρας του frame, αντισταθμίζονται με αυτή. Ο τύπος αποθήκευσής της, είναι (fixed point) `<s|16,13>`. Η τιμή της αποθηκεύεται σε έναν εσωτερικό καταχωρητή, μόλις σηκωθεί το `in_phi_v`. Η διαδικασία της αντιστάθμισης είναι συνεχής, όσο το `in_valid` είναι σηκωμένο. Ωστόσο υπάρχει ένα ελάχιστο delay 2 κύκλων από τη στιγμή που σηκώνεται το `in_phi_v`, μέχρι να αρχίζει να εφαρμόζεται η νέα τιμή για τη φάση.

Η είσοδος `in_n`, τύπου `<u|13>`, είναι ο χρονικός δείκτης του εκάστοτε μιγαδικού sample στις εισόδους `in_real`, `in_imag`. Ο δείκτης αυτός αντιστοιχεί στη μεταβλητή n της (4.1). Η τιμή του είναι μοναδική

¹⁶ Στη γενικότερη περίπτωση όπου η διαφορά φάσης είναι $\hat{\Phi} = \hat{\phi} + 2\pi\hat{z}$, αντιστάθμιση γίνεται με παρόμοιο τρόπο.

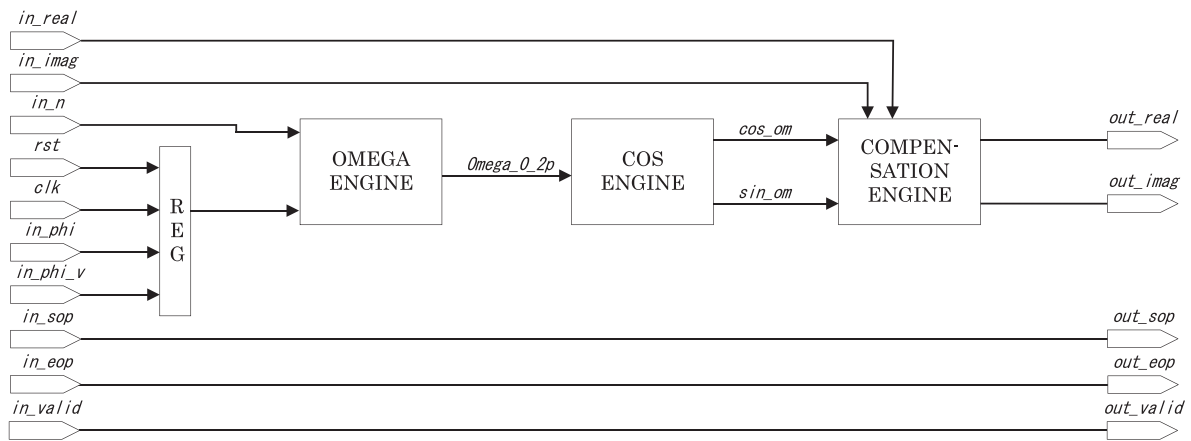
για κάθε sample ενός frame. Με το επιλεχθέν εύρος, υπάρχει δυνατότητα υποστήριξης frames ωφέλιμου μήκους $2^{13} = 8192$ samples. Το ωφέλιμο μήκος ενός frame εκφράζεται ως $RS = R_D N_s$ (samples), όπου $R_D = R - 3$ είναι το πλήθος των data symbols ανά frame¹⁷ και N_s το μήκος ενός OFDM συμβόλου.

Το ωφέλιμο μήκος ενός frame μετρημένο σε σύμβολα, εξαρτάται από την επιλογή του μεγέθους του OFDM συμβόλου, N_s . Για παράδειγμα, με την τηλεπικοινωνιακή σχεδίαση που παρουσιάστηκε στο Κεφάλαιο 2, ισχύει: $N_s = 160$, $R = 10 \Rightarrow RS = 1120$ samples. Εναλλακτικά, αν θεωρηθεί ως δεδομένο ότι $N_s = 160$, τότε $\max(R) = 54$ symbols.

Ανάλυση Μετά από ανάλυση του τύπου 4.1 προκύπτουν οι εξής εξισώσεις για τα αντισταθμισμένα samples:

$$\left. \begin{aligned} \text{re}_{r_c}(n) &= \text{re}_r(n) \times B(n, \hat{\phi}) + \text{im}_r(n) \times C(n, \hat{\phi}) \\ \text{im}_{r_c}(n) &= \text{im}_r(n) \times B(n, \hat{\phi}) - \text{re}_r(n) \times C(n, \hat{\phi}) \end{aligned} \right\} \quad (4.2)$$

όπου $B(n, \hat{\phi}) = \cos\left(\frac{2n}{N_o} \hat{\phi}\right)$ $C(n, \hat{\phi}) = \sin\left(\frac{2n}{N_o} \hat{\phi}\right)$



Σχήμα 4.15: FCO Top-Level διάγραμμα

Παρατηρώντας τις παραπάνω εξισώσεις διακρίνονται τα εξής 3 στάδια υπολογισμού :

1. Υπολογισμός του ορίσματος των τριγωνομετρικών συναρτήσεων $\psi = \frac{2n}{N_o} \hat{\phi}$
2. Υπολογισμός των τριγωνομετρικών συναρτήσεων $\cos(\psi)$, $\sin(\psi)$.
3. Υπολογισμός των μιγαδικών γινομένων αντιστάθμισης των samples εισόδου.

Το κύκλωμα που υλοποιείται υπολογίζει το κάθε ένα από αυτά τα στάδια ξεχωριστά. Στο Σχήμα 4.15 απεικονίζεται η top-level περιγραφή του κυκλώματος.

4.7.1 Υπολογιστής Ορίσματος - Omega engine

Ο υπολογιστής cos engine -που θα περιγραφεί στο επόμενο τμήμα- υπολογίζει τις $g_c(\Omega_c) = \cos(2\pi\Omega_c)$ και $g_s(\Omega_c) = \sin(2\pi\Omega_c)$ $\Omega_c \in [0, 1)$. Συνεπώς θα πρέπει να γίνει αναγωγή του εύρους της γωνίας ψ , στο

¹⁷Υπενθυμίζεται ότι υπάρχουν 3 training σύμβολα σε κάθε frame, τα οποία δε φέρουν ωφέλιμη πληροφορία. Το ωφέλιμο μήκος ενός frame, αναφέρεται στον αριθμό των data OFDM συμβόλων που ακολουθούν τα training σύμβολα. Δεν έχει καμία σχέση με το ωφέλιμο μήκος ενός OFDM συμβόλου.

εύρος της κανονικοποιημένης πρωτεύουσας γωνίας Ω_c .

Η αρχική γωνία είναι:

$$\psi = \frac{2n\hat{\phi}}{N_o} \quad \hat{\phi} \in [-\pi, \pi] \quad n \in [0, N_s * R_D - 1] \subseteq \mathbb{N} \quad (4.3)$$

και από την ιδιότητα της περιοδικότητας των $\cos(\psi)$, $\sin(\psi)$ ισχύει $\psi + 2\pi m = \psi$, $m \in \mathbb{Z}$. Εκφράζοντας την ψ συναρτήσει της κανονικοποιημένης γωνίας Ω_∞ ισχύει:

$$\psi = 2\pi\Omega_\infty \quad (4.4)$$

Επιπλέον για το μήκος του ωφέλιμου OFDM συμβόλου ισχύει $N_o = N$. Για το μέγεθος του FFT ισχύει $N = 2^u$, $u = \log_2(N)$, εφόσον οι επιτρεπόμενες τιμές του είναι δυνάμεις του 2. Συνεπώς μπορεί να γραφτεί:

$$N_o = 2^u \quad (4.5)$$

Από τις (4.3), (4.4) και (4.5) εξάγεται:

$$\Omega_\infty = \frac{\hat{\phi}n}{\pi 2^u} \quad \Omega_\infty \in (-\infty, +\infty) \quad (4.6)$$

Λόγω περιοδικότητας ισχύει $\Omega_\infty + m = \Omega_\infty$, $m \in \mathbb{Z}$. Από αυτή την ιδιότητα είναι σαφές πως η κανονικοποιημένη γωνία μπορεί να μειωθεί στον πρώτο κύκλο:

$$\Omega_c = \text{frac}(\Omega_\infty) \quad \Omega_c \in [0, 1)$$

όπου με $\text{frac}(x)$ συμβολίζεται το δεκαδικό μέρος της x .

Υλοποίηση

Για τον υπολογισμό της (4.6) γίνεται παραγοντοποίηση, ώστε να μειωθούν τα σφάλματα κατά τον υπολογισμό. Ο υπολογισμός γίνεται σύμφωνα με την:

$$\Omega_\infty = \underbrace{\left(\hat{\phi} \times \underbrace{\left(n \times \frac{1}{\pi} \right)}_A \right)}_B 2^{-u} \quad (4.7)$$

Η είσοδος $\hat{\phi}$ είναι τύπου $\langle s | 16 \cdot 13 \rangle$, όπως εξέρχεται από τον TSY, ενώ ο χρονικός δείκτης sample, n , είναι τύπου $\langle u | 13 \rangle$. Η σταθερά $\frac{1}{\pi}$ είναι προϋπολογισμένη και αποθηκεύεται σε $\langle u | 16 \cdot 16 \rangle$. Ο πολλαπλασιασμός με 2^{-u} υλοποιείται με ολίσηση u θέσεων προς τα δεξιά. Μετά από ανάλυση και δοκιμές ο τύπος εξόδου τέθηκε σε $\langle u | 15 \cdot 15 \rangle$, ενώ το εύρος τιμών που μπορεί να πάρει το u είναι $u \in \{6, 7, 8, 9, 10\} \Rightarrow N \in \{64, 128, 256, 512, 1024\}$. Υπενθυμίζεται ότι το N είναι σταθερό για ένα instance του κυκλώματος, αλλά μπορεί να αλλάξει εύκολα σε ένα διαφορετικό. Μετά από ανάλυση σφάλματος προέκυψε ότι, όσο μικραίνει το N τόσο μεγαλύτερη ακρίβεια πρέπει να έχει το μερικό γινόμενο B' στον (4.7). Συνεπώς, τέθηκε ως προδιαγραφή ένα κάτω όριο για το υποστηριζόμενο μέγεθος FFT, ώστε η ακρίβεια να είναι σε ανεκτά επίπεδα με κάθε επιλογή N .¹⁸ Οι εσωτερικές ακρίβειες εργασίας και η ροή του σήματος, απεικονίζονται στο datapath (Σχήμα 4.16).

¹⁸Το $\max(N)$ προέρχεται από τις προδιαγραφές του OFDM core.

4.7.2 Υπολογιστής cos/sin - Cos engine

Σε αυτό το στάδιο υπολογίζονται οι τριγωνομετρικές συναρτήσεις $g_c(\Omega_c) = \cos(2\pi\Omega_c)$ και $g_s(\Omega_c) = \sin(2\pi\Omega_c)$ $\Omega_c \in [0, 1)$, καθώς στο προηγούμενο στάδιο έχει υπολογιστεί το όρισμα τους, Ω_c . Η μέθοδος που επιλέχτηκε στο παρόν είναι υβριδικού τύπου, κάνοντας χρήση πολυωνυμικής και βασισμένης σε πίνακα προσέγγιση.

Η βασική αρχή των πολυωνυμικών μεθόδων, είναι πως μία συνάρτηση f μπορεί να προσεγγισθεί με την επιθυμητή ακρίβεια, σε ένα διάστημα ενδιαφέροντος του πεδίου ορισμού της, με ένα κατάλληλο πολυώνυμο n -βαθμού. Γενικά, ισχύει πως όσο μεγαλύτερος ο βαθμός n , τόσο ακριβέστερη η προσέγγιση. Ωστόσο, με την αύξηση του βαθμού, αυξάνονται οι πολλαπλασιασμοί και οι προσθέσεις, ενώ η ταχύτητα του υπολογισμού μειώνεται. Η ταχύτητα σύγκλισης της προσέγγισης, ως προς το βαθμό n , εξαρτάται πολύ από το είδος της συνάρτησης f .¹⁹

Για την εύρεση των κατάλληλων συντελεστών ενός πολυωνύμου, υπάρχουν δύο τρόποι:

- Προσεγγίσεις Ελαχίστου Τετραγώνου (*Least Squares Approximations*) και
- Προσεγγίσεις Ελαχίστου Μεγίστου (*Minimax Approximations*)

Οι πρώτες ελαχιστοποιούν το μέσο σφάλμα, ενώ οι δεύτερες το μέγιστο σφάλμα. Ενώ για τις προσεγγίσεις ελαχίστου τετραγώνου υπάρχουν αρκετοί αναλυτικοί προσδιορισμοί των συντελεστών του πολυωνύμου (Legendre, Jacobi, Chebychev, κ.α.), δε συμβαίνει το ίδιο και στις προσεγγίσεις minimax. Οι τελευταίες συνήθως υπολογίζονται με τη βοήθεια προγραμμάτων συμβολικής ανάλυσης.

Από την άλλη, στις table based μεθόδους, αποθηκεύονται προϋπολογισμένες τιμές της συνάρτησης σε σημεία του διαστήματος που ενδιαφέρει. Για την επίτευξη μεγαλύτερης ακρίβειας, χρειάζεται αυξημένη πυκνότητα των σημείων υπολογισμού. Αυτό με τη σειρά του, οδηγεί σε αυξημένες απαιτήσεις για το μέγεθος της μνήμης. Ωστόσο η ταχύτητα υπολογισμού είναι ανεξάρτητη της ακρίβειας. Με τις επιδόσεις των σημερινών μνημών, αλλά και με το χαμηλό κόστος τους, είναι δυνατή η επίτευξη ταχύτατων υπολογισμών μεγάλης ακρίβειας με table based προσεγγίσεις.²⁰

Στο παρόν υλοποιείται μία μέθοδος, που έχει προταθεί από τον Tang [Tan91], η οποία συνδυάζει τις δύο προηγούμενες. Το διάστημα ενδιαφέροντος τεμαχίζεται σε μικρότερα και η συνάρτηση προσεγγίζεται με πολυώνυμα μικρότερου βαθμού που αντιστοιχούν στο κάθε τμήμα. Οι συντελεστές των πολυωνύμων αποθηκεύονται σε μία μνήμη και εξαρτώνται από το εκάστοτε τμήμα. Με αυτόν τον τρόπο είναι δυνατό να επιτευχθεί μεγάλη ακρίβεια στην προσέγγιση, κρατώντας την υπολογιστική πολυπλοκότητα και το χρησιμοποιούμενη μνήμη σε χαμηλά επίπεδα.

Ο αλγόριθμος του Tang συνίσταται σε τρία βασικά βήματα (Με $f(x)$ συμβολίζεται η συνάρτηση της οποίας αναζητείται η προσέγγιση):²¹

A) Αναγωγή (Reduction) • Δεδομένου του x επιλέγεται ένα σημείο c_k , από ένα σύνολο προκαθορισμένων σημείων, ώστε να είναι πλησιέστερο στο x . Με αυτόν τον τρόπο καθορίζεται το τμήμα του διαστήματος ενδιαφέροντος, στο οποίο θα εφαρμοσθεί η προσέγγιση. Στη συνέχεια εφαρμόζεται ένας μετασχηματισμός αναγωγής αποκτώντας την ποσότητα $r = R(x, c_k)$.

B) Προσέγγιση (Approximation) • Υπολογίζεται η προσέγγιση της $f(r) \simeq p_k(r)$, όπου $p_k(x)$ είναι μια πολυωνυμική (συνήθως) προσέγγιση της $f(x)$ στο k -οστό διάστημα.

Γ) Ανακατασκευή (Reconstruction) • Η τελική τιμή της συνάρτησης ανακατασκευάζεται με τη βοήθεια ενός μετασχηματισμού ώστε να ισχύει: $f(x) = S(f(c_k), p_k(r))$.

¹⁹Για μία εκτενή συζήτηση πάνω στις πολυωνυμικές προσεγγίσεις βλ. [Mul06, Κεφ. 3]

²⁰Για μία εκτενή συζήτηση πάνω στις προσεγγίσεις βασισμένες σε πίνακα [Mul06, Κεφ. 4]

²¹Εδώ παρουσιάζεται μία ελαφρώς διαφορετική περιγραφή του αλγορίθμου του Tang, καθώς περιγράφει την παρούσα υλοποίηση. Συγκεκριμένα, στην αρχική τοποθέτηση του Tang, το βήμα B βασίζεται σε ένα πολυώνυμο $p(r)$ και όχι σε ανά τμήμα πολυώνυμο $p_k(r)$. Ωστόσο η λογική παραμένει η ίδια.

Επιθυμητή ακρίβεια

Για τον υπολογισμό της $f(x)$, το διάστημα ενδιαφέροντος χωρίζεται σε τμήματα όπου στο καθένα γίνεται πολυωνυμική προσέγγιση. Ένα σημείο που πρέπει να ξεκαθαριστεί, είναι η επιλογή του πλήθους των τμημάτων και του βαθμού του πολυωνύμου. Αυτή θα γίνει σύμφωνα με την επιθυμητή ακρίβεια της προσέγγισης και τις απαιτήσεις σε υλικό/απόδοση. Ο προσδιορισμός της επιθυμητής ακρίβειας προϋποθέτει λεπτομερή ανάλυση των σημάτων που εμπλέκονται στους υπολογισμούς. Ωστόσο η επίδοση ενός τηλεπικοινωνιακού συστήματος μετράται κυρίως από το BER ή το ISNR. Ο προσδιορισμός αναλυτικής σχέσης που συσχετίζει την ακρίβεια υπολογισμού του cos engine, με το ISNR είναι πέρα από τους σκοπούς του παρόντος.

Μία εναλλακτική διαδικασία είναι η επιλογή κάποιων τιμών, βασισμένων σε μία στοιχειώδη ανάλυση και η επαλήθευσή τους κατά το στάδιο της αξιολόγησης τους συστήματος (system evaluation). Αν δεν ικανοποιούνται οι προδιαγραφές του συστήματος, τότε μπορεί να γίνει επανάληψη της διαδικασίας, επιλέγοντας διαφορετική ακρίβεια. Αυτή είναι η στρατηγική που ακολουθήθηκε στο παρόν.

Υλοποίηση

Ο σκοπός, εδώ είναι ο υπολογισμός των συναρτήσεων $g_c(\Omega_c) = \cos(2\pi\Omega_c)$ και $g_s(\Omega_c) = \sin(2\pi\Omega_c)$ $\Omega_c \in [0, 1)$. Η μηχανή (cos engine) που υλοποιήθηκε, παράγει ταυτόχρονα το $g_c(\Omega_c)$ και $g_s(\Omega_c)$ και βασίστηκε στο [Lee+06].²² Μία περιγραφή του αλγορίθμου του tang που εφαρμόστηκε εδώ, αναγράφεται στη Λίστα 4.1.

```

1  -- TANG'S ALGORITHM PSEUDOCODE FOR COS ENGINE --
2  ----- Evaluate g_s=sin(2*pi*Om) -----
3  -----          g_c=cos(2*pi*Om) -----
4
5  -- Range Reduction
6  quad = Om[14:13];
7  x1 = Om[12:0];
8  x2 = (1-2^-13)-x1;
9
10 -- Approximate cos(x_1*pi/2) and cos(x_2*pi/2)
11 -- Degree-1 piecewise polynomial
12 ya = C1[x1]*x1 + C0[x1];
13 yb = C1[x2]*x2 + C0[x2];
14
15 -- Range Reconstruction
16 switch (quad)
17 case 0: g_s = yb; g_c = ya;
18 case 1: g_s = ya; g_c = -yb;
19 case 2: g_s = -yb; g_c = -ya;
20 case 3: g_s = -ya; g_c = yb;

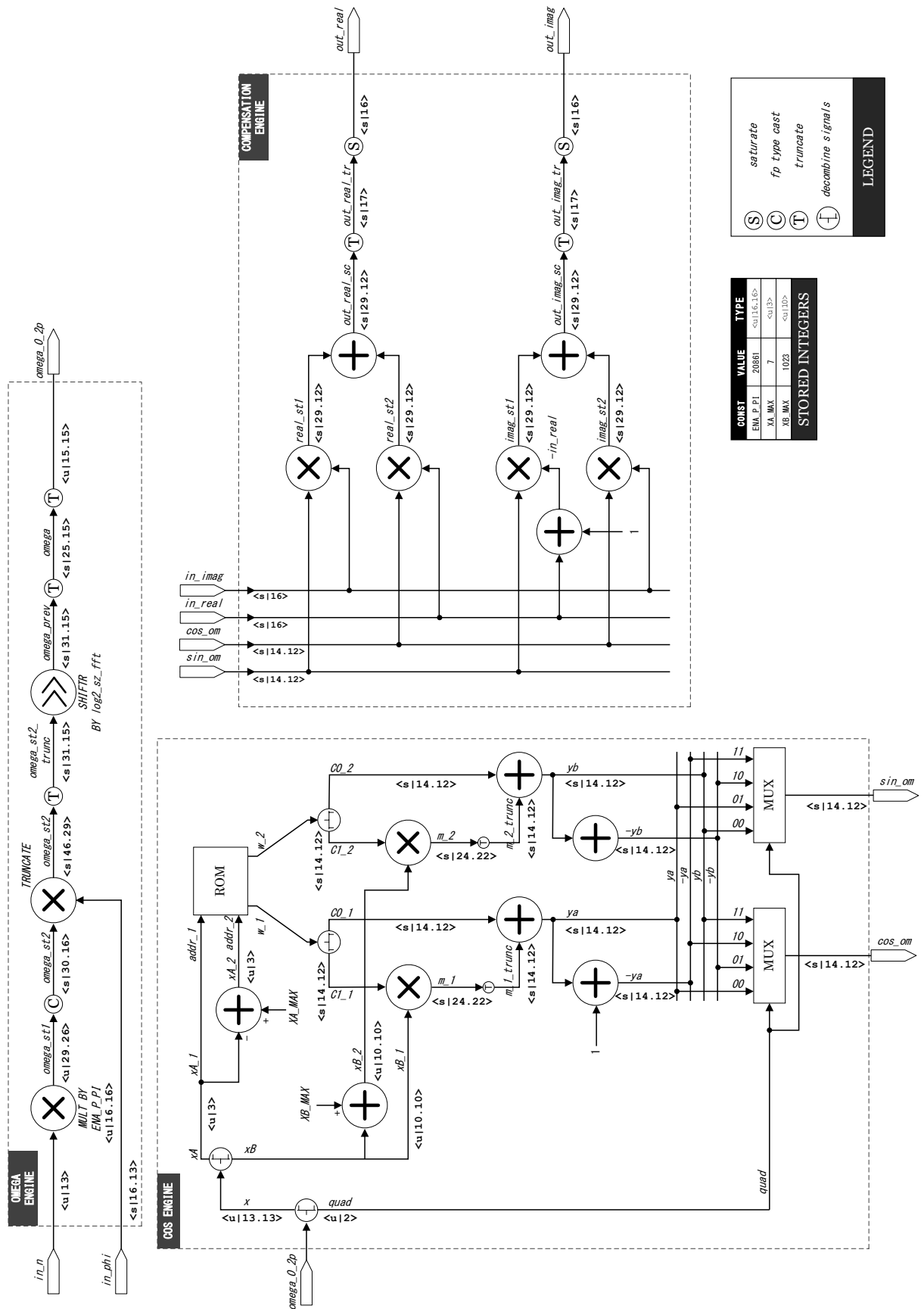
```

Λίστα 4.1: Ψευδοκώδικας με τον αλγόριθμο του tang για το cos engine

Για λόγους συντομίας στις επεξηγήσεις που ακολουθούν, για την αριθμητική ποσότητα y που αναπαρίσταται με κάποιο τύπο fixed point $\langle s | B\{y\} \cdot F\{y\} \rangle$, συμβολίζεται με $B\{y\}$ το συνολικό εύρος της λέξης και με $F\{y\}$ το εύρος του δεκαδικού της μέρους.²³

²²Θα μπορούσε να είχε υλοποιηθεί η $\cos(\phi)$ και $\sin(\phi)$, $\phi \in [0, 2\pi)$, αλλά η επιλογή του πεδίου ορισμού στο διάστημα $[0, 1)$ απλοποιεί αρκετά το στάδιο αναγωγής και της ανάλυσης σφάλματος.

²³Για μια σύντομη περιγραφή του τύπου fixed point και του σχετικού συμβολισμού βλ. §4.5.2.



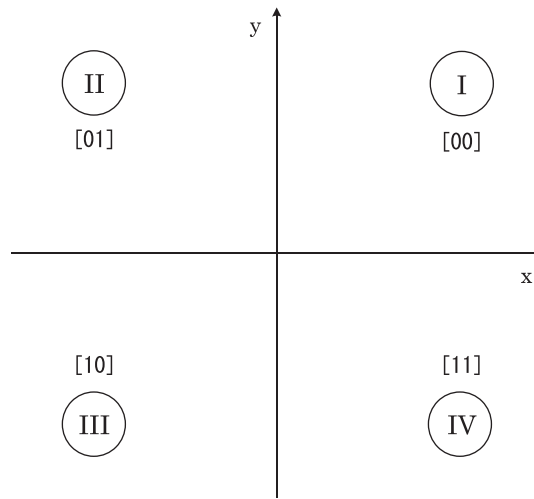
Σχήμα 4.16: Υπολογιστικές μηχανές του FCO

Αναγωγή

Πρώτη Αναγωγή Εδώ γίνεται χρήση των ιδιοτήτων συμμετρίας και περιοδικότητας των \cos/\sin με αποτέλεσμα την αναγωγή της προσέγγισης και των δύο συναρτήσεων, $\cos(\psi)$, $\sin(\psi)$, $\psi \in [0, 2\pi)$, στο πρώτο τεταρτημόριο της $\cos(\phi)$, $\phi \in [0, \frac{\pi}{2})$. Οι αναγωγικοί τύποι φαίνονται στον Πίνακα 4.11 ενώ η αρίθμηση των τεταρτημορίων στο Σχήμα 4.17.

Τετ/ριο	$\cos(\psi)$	$\sin(\psi)$
<i>I</i>	$\cos(\phi)$	$\sin(\phi) = \cos(\frac{\pi}{2} - \phi)$
<i>II</i>	$-\cos(\phi)$	$\sin(\phi) = \cos(\frac{\pi}{2} - \phi)$
<i>III</i>	$-\cos(\phi)$	$-\sin(\phi) = -\cos(\frac{\pi}{2} - \phi)$
<i>IV</i>	$\cos(\phi)$	$-\sin(\phi) = -\cos(\frac{\pi}{2} - \phi)$

Πίνακας 4.11: Αναγωγικοί τύποι των $\cos(\psi)$, $\sin(\psi)$ στο πρώτο τεταρτημόριο



Σχήμα 4.17: Αρίθμηση τεταρτημορίων για το \cos engine

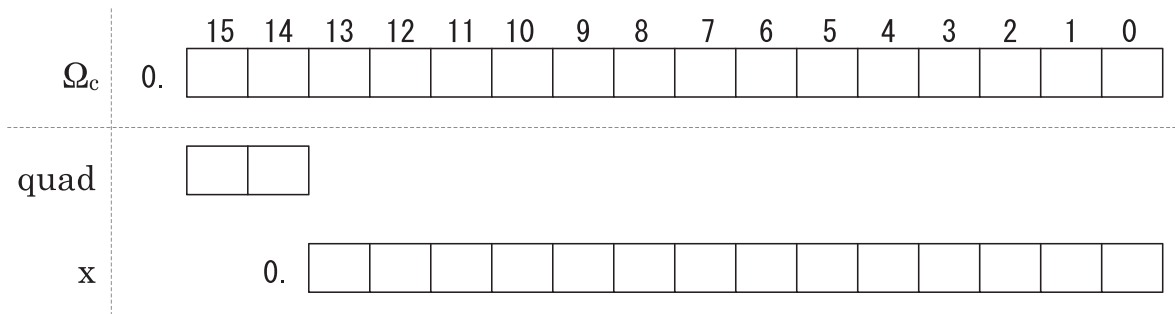
Δηλαδή οι $g_c(\Omega_c)$ και $g_s(\Omega_c)$, $\Omega_c \in [0, 1)$ μπορούν να ανακατασκευαστούν από τις²⁴

$$\left. \begin{aligned} ya(x) &= \cos\left(\frac{\pi}{2}x\right) \\ yb(x) &= \cos\left(\frac{\pi}{2}(1-x)\right) = \sin\left(\frac{\pi}{2}x\right) \\ x &\in [0, 1) \end{aligned} \right\} \quad (4.8)$$

Το πρώτο στάδιο αναγωγής, συνίσταται στην εξαγωγή της πληροφορίας του τεταρτημορίου και υλοποιείται κατατμώντας τη λέξη Ω_c σε υπο-λέξεις. Το τεταρτημόριο κωδικοποιείται με τα 2 MSB της Ω_c και εκχωρείται στη μεταβλητή *quad*. Τα υπόλοιπα εκχωρούνται στη μεταβλητή x και περνάνε στο δεύτερο στάδιο προσέγγισης της $ya(x)$, $yb(x)$. Ισχύει $x \in [0, 1)$. Η κατάτμηση της Ω_c σε υπολέξεις απεικονίζεται παραστατικά στο σχήμα 4.18, ενώ οι τιμές της *quad* για τα διάφορα τεταρτημόρια απεικονίζονται μέσα σε αγκύλες στο Σχήμα 4.17.

Τάξη πολυωνύμου Για την παρούσα υλοποίηση έχει επιλεγθεί πολυώνυμο πρώτης τάξης. Αυτό οδηγεί σε απλή και γρήγορη υλοποίηση που περιλαμβάνει ένα MULT και έναν ADD για το στάδιο της προσέγγισης.

²⁴ Αυτό προκύπτει, αν γίνουν οι αντικαταστάσεις $\psi = 2\pi\Omega_c$ και $\phi = \frac{\pi}{2}x$.

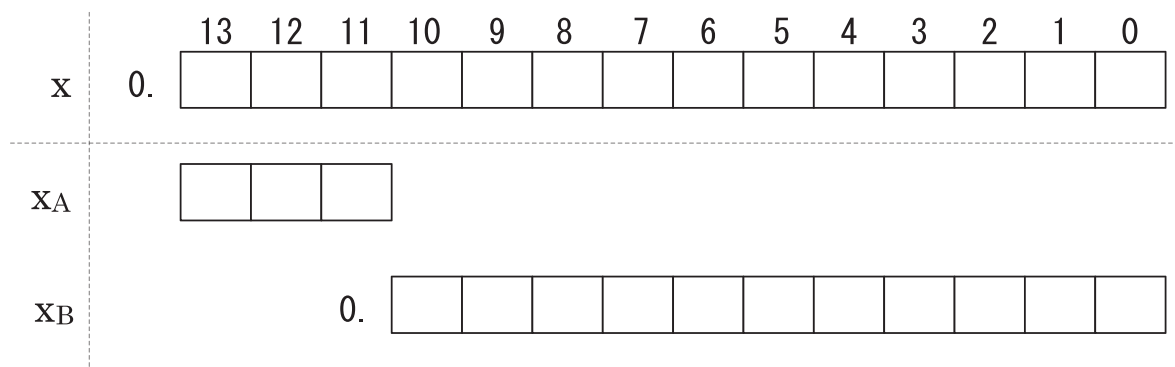
Σχήμα 4.18: Κατάτμηση της Ω_c σε υπολέξεις

Τμήμα (x_A)	$\varphi \in \dots$	$x \in \dots$
0	$[0, \frac{\pi}{16})$	$[0, \frac{1}{8})$
1	$[\frac{\pi}{16}, \frac{2\pi}{16})$	$[\frac{1}{8}, \frac{2}{8})$
2	$[\frac{2\pi}{16}, \frac{3\pi}{16})$	$[\frac{2}{8}, \frac{3}{8})$
3	$[\frac{3\pi}{16}, \frac{4\pi}{16})$	$[\frac{3}{8}, \frac{4}{8})$
4	$[\frac{4\pi}{16}, \frac{5\pi}{16})$	$[\frac{4}{8}, \frac{5}{8})$
5	$[\frac{5\pi}{16}, \frac{6\pi}{16})$	$[\frac{5}{8}, \frac{6}{8})$
6	$[\frac{6\pi}{16}, \frac{7\pi}{16})$	$[\frac{6}{8}, \frac{7}{8})$
7	$[\frac{7\pi}{16}, \frac{8\pi}{16})$	$[\frac{7}{8}, \frac{8}{8})$

Πίνακας 4.12: Διαστήματα προσέγγισης των cos/sin

Η γενική μορφή του πολωνύμου είναι:

$$p_k(x) = C_{1,k} \times x + C_{0,k} \quad (4.9)$$

Σχήμα 4.19: Κατάτμηση της x σε υπολέξεις

Δεύτερη Αναγωγή Αυτό το στάδιο, συνίσταται στην επιλογή του τμήματος k και στον υπολογισμό της ανεξάρτητης μεταβλητής του πολωνύμου. Η πιο απλή υλοποίηση, είναι να θεωρηθούν τα $B\{x_A\}$ πιο σημαντικά bits της λέξης x ως η επιλογή τμήματος και να αποθηκευτούν στη μεταβλητή x_A . Τα εναπομείναντα $B\{x_B\}$ bits, αποθηκεύονται στη μεταβλητή x_B και αποτελούν το r στον αλγόριθμο του Tang. Η κατάτμηση της λέξης x στις υπο-λέξεις x_A και x_B , απεικονίζεται παραστατικά στο Σχήμα 4.19. Απόρροια των παραπάνω είναι ότι το πλήθος των τμημάτων ισούται με $N_{seg} = 2^{B\{x_A\}}$. Στην παρούσα υλοποίηση επιλέχθηκε $B\{x_A\} = 3 \Rightarrow N_{seg} = 8$. Τα διαστήματα των τμημάτων αποτυπώνονται στον Πίνακα 4.12.

Προσέγγιση

Για τον εύρεση του βέλτιστου πολυωνύμου σε κάθε διάστημα εφαρμόστηκε το κριτήριο minimax, το οποίο ελαχιστοποιεί το μέγιστο σφάλμα προσέγγισης.²⁵ Καθώς η εφαρμογή του περιλαμβάνει εξαντλητική αναζήτηση, έγινε χρήση του λογισμικού Maple, με τη βοήθεια του οποίου υπολογίστηκαν οι συντελεστές των πολυωνύμων C_1, C_0 . Οι τιμές τους για κάθε διάστημα, καθώς και τα αντίστοιχα σφάλματα προσέγγισης, αναγράφονται στον πίνακα 4.12. Όπως φαίνεται, το μέγιστο σφάλμα προσέγγισης είναι $\max\{E_{approx}\} = 0.0024$. Ο κώδικας στο Maple για τον υπολογισμό τους αναγράφεται στη Λίστα 4.2.

```
with(numapprox);
Digits := 12;
for i from 1 to 8
do
i, minimax( cos( ( pi / 2 ) * x ), x = ( i - 1 ) * 1 / 8 .. i * 1 / 8, [ 1, 0 ], 1, 'err' ),
err;
od;
```

Λίστα 4.2: Κώδικας Maple για τον υπολογισμό των συντελεστών

Τμήμα	C_0	C_1	E_{approx}
0	1.00239604907	-0.153717756776	0.002396049844310
1	1.03999504613	-0.455245983136	0.002304024938800
2	1.11082287112	-0.739279361664	0.002123518480630
3	1.20641954842	-0.994902648936	0.001861484699840
4	1.31478093106	-1.212292385340	0.001528033012010
5	1.42114018358	-1.383094405260	0.001136079671390
6	1.50894289058	-1.500744882800	0.000701046862808
7	1.56096475712	-1.560722576100	0.000242181023078

Πίνακας 4.13: Συντελεστές πολυωνύμων 1^{ης} τάξης της $\cos(\frac{\pi}{2}x)$ στο διάστημα $[0, 1)$

Μετασχηματισμοί

Όπως προαναφέρθηκε, οι συντελεστές που υπολογίστηκαν προσεγγίζουν τη συνάρτηση $\cos(\frac{\pi}{2}x)$, $x \in [0, 1)$. Εφόσον η επιλογή του τμήματος γίνεται από τα 3 MSB της x , που αντιστοιχούν στη x_A , είναι φανερό πως μπορούν να απαλειφθούν από τους υπολογισμούς. Με αυτόν τον τρόπο επιτυγχάνεται μείωση του εύρους των απαιτούμενων πράξεων για την προσέγγιση. Για το λόγο αυτό επιλέγεται η x_B ως η μεταβλητή του πολυωνύμου. Για να εξακολουθεί να υπάρχει μαθηματική συνέπεια, οι συντελεστές θα πρέπει να μετασχηματιστούν.

Η επιλογή της μεταβλητής x_B περιγράφεται λειτουργικά ως εξής: Στη μεταβλητή x μηδενίζονται τα 3 MSB, αφού εκχωρηθούν στη μεταβλητή x_A και εκτελείται ολίσθηση 3 θέσεων προς τα αριστερά. Το αποτέλεσμα εκχωρείται στην x_B . Η ίδια διαδικασία περιγράφεται μαθηματικά ως:

$$x_B = (x - x_A) 2^{B\{x_A\}}$$

Επιλύοντας την παραπάνω σχέση, ως προς x και αντικαθιστώντας στην (4.9) αποκτάται το ακόλουθο ζεύγος μετασχηματισμών :

$$\hat{C}_{1,k} = \frac{C_{1,k}}{2^{B\{x_A\}}}$$

²⁵Για το κριτήριο minimax έχει ήδη γίνει αναφορά στην §4.6.6.

$$\hat{C}_{0,k} = C_{1,k} x_{A,k} + C_0$$

Αποθήκευση

Το τελικό βήμα είναι η αποθήκευση των συντελεστών στη μνήμη. Μετά από ανάλυση σφάλματος και δοκιμές, επιλέχθηκε ο τύπος <s | 14 . 12>. Για την αποθήκευση σε αυτόν τον τύπο, οι υπολογισμένοι συντελεστές στρογγυλοποιούνται στον κοντινότερο αριθμό (round) με τη βοήθεια προγράμματος MATLAB. Το μέγιστο σφάλμα στρογγυλοποίησης είναι 0.000119, αρκετά μικρότερο του σφάλματος προσέγγισης.

Όπως έχει αναφερθεί στην §4.5.2, οι πράξεις μεταξύ fixed point αριθμών μπορούν να γίνουν από ένα συμβατικό κύκλωμα που χειρίζεται ακεραίους. Αυτή είναι η προσέγγιση που ακολουθείται στην παρούσα υλοποίηση. Δηλαδή οι πράξεις γίνονται μεταξύ ακεραίων, ενώ αυτοί ερμηνεύονται ως fixed point όπου αυτό έχει νόημα.²⁶ Για το λόγο αυτό είναι χρήσιμος ο αποθηκευμένος ακέραιος που αντιστοιχεί στην fixed point αναπαράσταση μιας σταθεράς.

Οι τελικοί, μετασχηματισμένοι και στρογγυλοποιημένοι συντελεστές, καθώς και οι αποθηκευμένοι ακέραιοι που τους αντιστοιχούν, αναγράφονται στον Πίνακα 4.14.

Τμήμα	\hat{C}_0	\hat{C}_1	$I\{\hat{C}_0\}$	$I\{\hat{C}_1\}$
0	1.002441406250	-0.019287109375	4106	-79
1	0.983154296875	-0.056884765625	4027	-233
2	0.926025390625	-0.092529296875	3793	-379
3	0.833251953125	-0.124267578125	3413	-509
4	0.708740234375	-0.151611328125	2903	-621
5	0.556640625000	-0.172851562500	2280	-708
6	0.383300781250	-0.187500000000	1570	-768
7	0.195312500000	-0.195068359375	800	-799

Πίνακας 4.14: Τελικοί συντελεστές \hat{C}_1 , \hat{C}_0 και οι αντίστοιχοι stored integers

Ανακατασκευή

Μέχρι αυτό το σημείο έχει υπολογιστεί η τιμή της $ya(x) = \cos(\frac{\pi}{2}x)$ και της $yb(x) = \sin(\frac{\pi}{2}x)$ με τη γωνία του ορίσματος να βρίσκεται στο πρώτο τεταρτημόριο, $x \in [0, 1)$. Για την ανακατασκευή και στα υπόλοιπα τεταρτημόρια θα πρέπει να ληφθεί υπόψη η μαθηματική σχέση μεταξύ x και Ω_c . Το πρώτο στάδιο αναγωγής θα μπορούσε να περιγραφεί ως εξής: Αρχικά μηδενίζονται τα 2 MSB της Ω_c αφού έχουν εκχωρηθεί στην *quad*. Στη συνέχεια γίνεται ολίσθηση δύο θέσεων προς τα αριστερά. Το αποτέλεσμα αποθηκεύεται στη μεταβλητή x . Συνεπώς ισχύει: $x = (\Omega_c - quad) 2^2$. Επιλύοντας ως προς Ω_c και αντικαθιστώντας στις $g_c(\Omega_c)$ και $g_s(\Omega_c)$ αποκτάται:

$$g_c(\Omega_c) = \cos\left(\pi\left(\frac{x}{2} + 2quad\right)\right)$$

$$g_s(\Omega_c) = \sin\left(\pi\left(\frac{x}{2} + 2quad\right)\right)$$

Έξοδοι του cos engine Εφαρμόζοντας τις γνωστές τριγωνομετρικές ταυτότητες και κάνοντας χρήση των εξισώσεων (4.8), οι $g_c(\Omega_c)$, $g_s(\Omega_c)$ εκφράζονται συναρτήσει των $ya(x)$, $yb(x)$ σύμφωνα με τον Πίνακα 4.15. Σημειώνεται πως αυτές είναι και οι τελικές έξοδοι του cos engine.

²⁶ Αυτό αποτυπώνεται στον κώδικα VHDL χρησιμοποιώντας τα κλασικά πακέτα αριθμητικής integer και όχι κάποια βιβλιοθήκη fixed point αριθμητικής.

quad	g_c	g_s
'00'	ya	yb
'01'	$-yb$	ya
'10'	$-ya$	$-yb$
'11'	yb	$-ya$

Πίνακας 4.15: Τύποι ανακατασκευής των $g_c(\Omega_c)$, $g_s(\Omega_c)$

Ανάλυση ακρίβειας συντελεστών

Τα στάδια της αναγωγής και της ανακατασκευής δεν εισάγουν σφάλμα. Αντιθέτως το στάδιο της προσέγγισης εισάγει σφάλμα και είναι αντικείμενο ανάλυσης η επιλογή των ενδιάμεσων ακριβειών. Η διαδικασία που μπορεί να ακολουθηθεί είναι αυτή της τεχνικής mini-bit, όπως περιγράφεται στο [Lee+06]. Η ανάλυση που ακολουθεί για το ya εφαρμόζεται και στην περίπτωση του yb . Η ακρίβεια των μεταβλητών εξόδου g_c , g_s , είναι ίδια με αυτή των ya , yb .

Η πράξη που εκτελείται είναι:

$$ya = \hat{C}_0 + (\hat{C}_1 \times x_B) = C'_0 + D_0$$

Το πρόβλημα είναι η επιλογή των δεκαδικών ψηφίων για τις μεταβλητές \hat{C}_0 , \hat{C}_1 , D_0 . Από την απαίτηση το σφάλμα στρογγυλοποίησης E_{ya} να είναι μικρότερο από το μέγιστο σφάλμα προσέγγισης E_{approx} , προκύπτει $B\{ya\} = 9$, αφού ισχύει

$$E_{ya} = 2^{-9} \leq 2.4 \cdot 10^{-3}$$

Το σφάλμα αυτό κρίνεται ικανοποιητικό για το στάδιο της αντιστάθμισης συχνότητας που ακολουθεί, όπως προκύπτει από δοκιμές που έγιναν στο κύκλωμα. Η ανάλυση mini-bit, του σφάλματος E_{ya} δίνει :

$$\begin{aligned} E_{ya} &= E_{\hat{C}_0} + E_{D_0} + 2^{-9} + E_{approx} \\ E_{D_0} &= \max\{x_B\} E_{\hat{C}_1} + \max\{\hat{C}_1\} E_{x_B} + 2^{-F\{D_0\}} \\ E_{\hat{C}_0} &= 2^{-F\{\hat{C}_0\}}, \quad E_{\hat{C}_1} = 2^{-F\{\hat{C}_1\}}, \quad \max\{x_B\} = 1 \\ \max\{\hat{C}_1\} &= 1, \quad E_{x_B} = 2^{-10} \end{aligned}$$

Κάνοντας τις κατάλληλες αντικαταστάσεις αποκτάται :

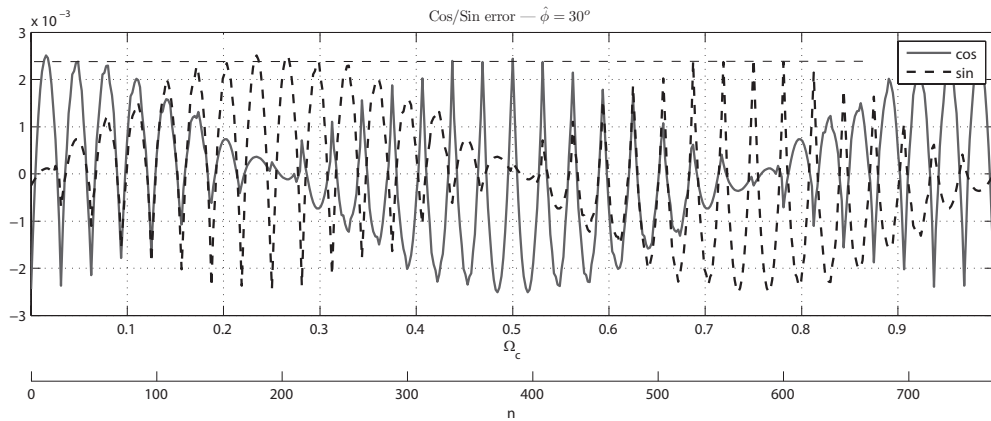
$$2^{-F\{D_0\}} + 2^{-F\{\hat{C}_0\}} + 2^{-F\{\hat{C}_1\}} \leq 4.977 \cdot 10^{-3}$$

Η παραπάνω εξίσωση έχει αρκετές λύσεις. Η επιλογή που έγινε, είναι :

$$F\{\hat{C}_0\} = F\{\hat{C}_1\} = F\{D_0\} = 12 \Rightarrow 0.733 \cdot 10^{-3} \leq 4.977 \cdot 10^{-3}$$

Η επιλογή όλων των ενδιάμεσων και τελικών ακριβειών απεικονίζεται στο datapath του κυκλώματος (Σχήμα 4.16). Στο Σχήμα 4.20 απεικονίζεται το σφάλμα του συνδυασμού cos engine και Omega computer για έναν πλήρη κύκλο της Ω_c , όπου έχει τεθεί $\hat{\phi} = 30^\circ$.

Για την αποτίμηση της ακρίβειας των υπολογισμών δημιουργήθηκε μοντέλο του κυκλώματος στο MATLAB, κάνοντας χρήση του *Fixed Point Toolbox*. Οι πράξεις εκτελέστηκαν με τρόπο παρόμοιο με το κύκλωμα. Με αυτόν τον τρόπο επιλέχθηκαν οι κατάλληλες ακρίβειες εργασίας και υπολογίστηκε το σφάλμα για το συνδυασμό omega και cos engine. Στη συνέχεια οι ακρίβειες εργασίας μεταγράφηκαν στον κώδικα VHDL που υλοποιεί το κύκλωμα.



Σχήμα 4.20: Σφάλμα του cos engine για έναν πλήρη κύκλο της Ω_c

4.7.3 Αντιστάθμιση - Compensation engine

Η αντιστάθμιση επιτελείται υλοποιώντας απευθείας τις εξισώσεις (4.2), κάνοντας τις εξής εκχωρήσεις σημάτων :

$$\begin{array}{ll}
 B(n, \hat{\phi}) \rightarrow ya & C(n, \hat{\phi}) \rightarrow yb \\
 re_r(n) \rightarrow in_real & im_r(n) \rightarrow in_imag \\
 re_{rc}(n) \rightarrow out_real & im_{rc}(n) \rightarrow out_imag
 \end{array}$$

Η επιλογή των ενδιάμεσων ακριβειών και ο ροή του σήματος, απεικονίζονται στο datapath του κυκλώματος (Σχήμα 4.16)

Εδώ γίνεται υπολογισμός του κάθε στελέχους της εξόδου με πλήρη ακρίβεια (12 δεκαδικά bit) μέχρι την έξοδο των adder. Καθώς ο τύπος εξόδου είναι $\langle s | 16 \rangle$ γίνεται στη συνέχεια αποκοπή των δεκαδικών bit, οδηγώντας σε $\langle s | 17 \rangle$. Τέλος γίνεται μετατροπή στον τελικό τύπο με κορεσμό, ώστε να αποφευχθούν φαινόμενα αναδίπλωσης.

4.8 Fine Frequency Estimator [FFE]

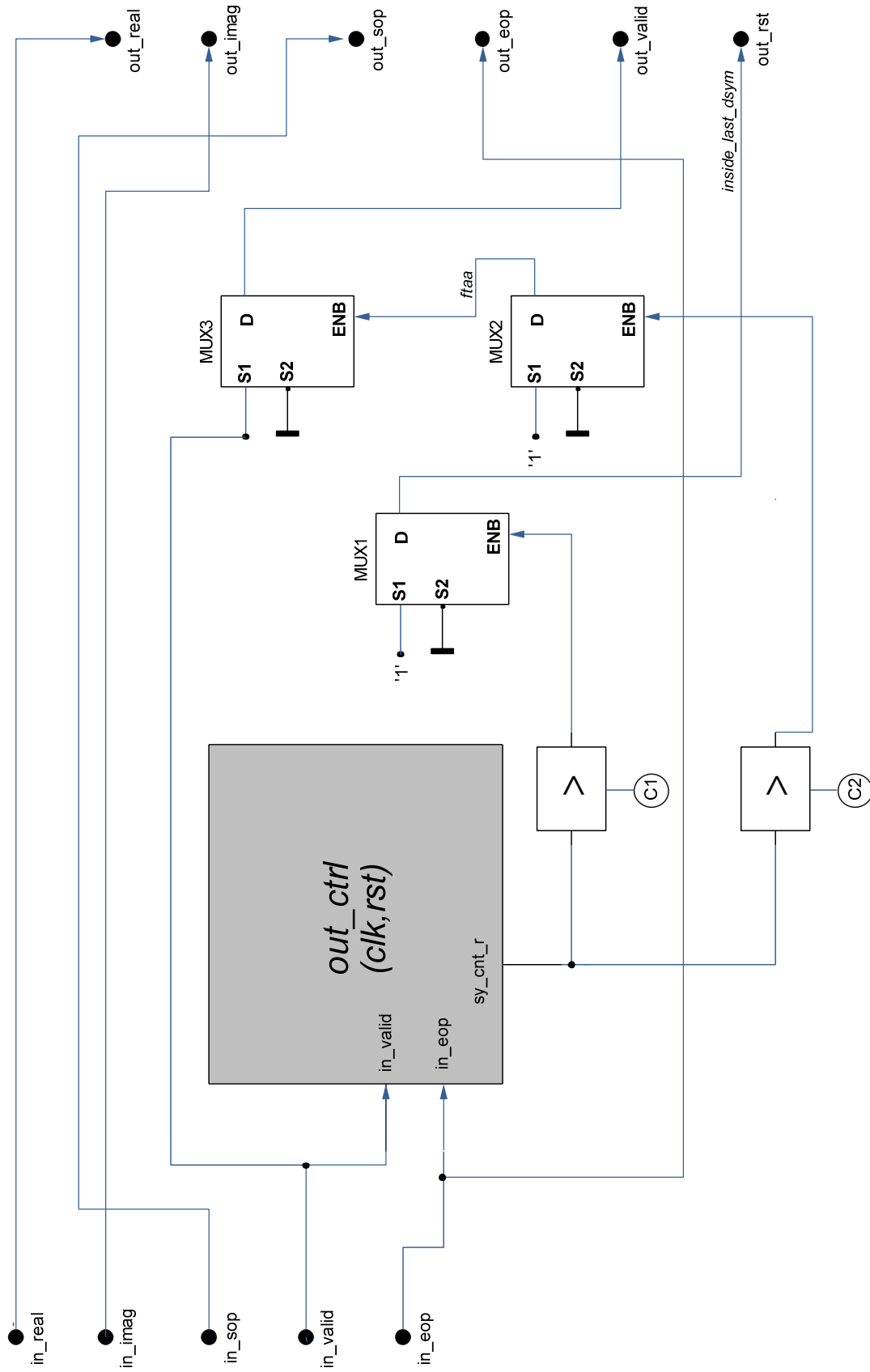
4.8.1 Η Top-Level Όψη

Στο σχήμα 4.21 παρουσιάζεται το top-level διάγραμμα του fine frequency estimator. Στην τρέχουσα υλοποίηση σκοπός του είναι η αφαίρεση των δυο training συμβόλων, που έχουν επιτελέσει πλέον το ρόλο τους, από το frame. Λόγω αυτού, είναι ένα module το οποίο επιβάλλει frontpressure στην downstream αλυσίδα.

4.8.2 Περιγραφή Κυκλωματικής Λειτουργίας

Το υποσύστημα αυτό είναι "διάφανο" ως προς τα σήματα out_real , out_imag , out_sop , out_eop με την έννοια ότι δεν υφίστανται κανενός είδους επεξεργασία, ούτε και χρονική καθυστέρηση μέσω μανδαλωτών.

Πυρήνας του υποσυστήματος είναι η διαδικασία μέτρησης (counter process) η οποία δέχεται ως άτυπες εισόδους τα σήματα in_valid , in_eop και παρέχει ως άτυπη έξοδο το σήμα cnt_s_r . Το σήμα αυτό περιέχει



Σχήμα 4.21: FFE Top-Level διάγραμμα

τον "αύξοντα αριθμό" του κάθε symbol. Το σήμα *in_valid* χρησιμοποιείται για τη διακοπή της διαδικασίας μέτρησης (stall) σε περίπτωση που τα δεδομένα εισόδου δεν είναι έγκυρα (*in_valid* = '0'), ενώ η συνθήκη *in_eor*='1' οδηγεί στην αύξηση του *cnt_s_r* κατά 1 κάθε φορά.

Τώρα, δύο βασικά εσωτερικά σήματα αποτελούν τα *inside_last_dsym* και *from_third_and_after* (στο διάγραμμα αναφέρεται ως *ftaa*) τα οποία είναι έξοδοι των πολυπλεκτών MUX-1 και MUX-2, αντίστοιχα. Πιο συγκεκριμένα :

MUX-1

Το σήμα εξόδου είναι το *inside_last_dsym* και υποδηλώνει ότι στο module εισέρχεται το τελευταίο data symbol του frame. Το σήμα επιλογής προκύπτει μετά από κατάλληλη σύγκριση, όπως φαίνεται παρακάτω.

```
1 -- Flag the last data symbol of the frame
2 inside_last_dsym <= '1' when (sy_cnt_r = (SZ_FRAME-1)) else '0';
```

MUX-2

Το άλλο σήμα (*from_third_and_after*), προέρχεται από τον MUX-2 ως εξής :

```
1 -- Flag the channel estim/comp symbols
2 from_third_and_after <= '1' when (sy_cnt_r > 1) else '0';
```

και σηματοδοτεί την απαρχή του ωφέλιμου frame (το τμήμα του frame που πρέπει να περάσει στο downstream module), δηλαδή, το εναπομείναν training symbol καθώς και τα data symbol.

MUX-3

Το παραπάνω σήμα χρησιμοποιείται σαν επιλογέας στον τρίτο πολυπλέκτη. Αυτός ο πολυπλέκτης υλοποιεί το μηχανισμό frontpressure που επιβάλλει αυτό το module στο downstream, καθορίζοντας την τιμή του σήματος *out_valid* ως εξής:

```
1 out_valid <= in_valid when from_third_and_after = '1' else '0';
```

Στην πραγματικότητα, λοιπόν, η "αφαίρεση" των δυο πρώτων training symbol, γίνεται με τη βοήθεια του σήματος *out_valid*, το οποίο είναι κρατημένο στο λογικό '0' κατά τη διάρκεια των πρώτων training symbol ενώ γίνεται '1' στα υπόλοιπα σύμβολα του frame.

4.9 Channel Estimator/Equalizer [CEQ]

Ο CEQ πραγματοποιεί την εκτίμηση, καθώς και την ισοστάθμιση καναλιού. Όπως έχει αναλυθεί στο κεφάλαιο 2, η εκτίμηση καναλιού γίνεται με τον αλγόριθμο LS. Για την πραγματοποίησή της, μεταδίδεται ένα training block (symbol) μία φορά ανά frame, μετά τα training symbol του συγχρονισμού. Η διαδικασία της εκτίμησης, που πραγματοποιείται κατά τη διάρκεια μετάδοσης του training block, έχει ως αποτέλεσμα τον υπολογισμό των συντελεστών του καναλιού για κάθε subcarrier, $H_e(k)$ $k \in [0, N_{SC} - 1]$. Εναλλακτικά η εκτίμηση του καναλιού μπορεί να συμβολιστεί με το διάνυσμα $\vec{H}e$. Στη συνέχεια τα data symbols που ακολουθούν το training block μέχρι το πέρας του frame, ισοσταθμίζονται με το $\vec{H}e$. Δηλαδή η ίδια εκτίμηση χρησιμοποιείται τόσες φορές όσες και το πλήθος των data symbols. Η διαδικασία εκτίμησης-ισοστάθμισης, επαναλαμβάνεται σε κάθε OFDM frame.

Διατηρώντας τους συμβολισμούς της §2.2.5, η εκτίμηση καναλιού υπολογίζεται ως:

$$He(k) = \frac{1}{\hat{H}_{LS}(k)} = \frac{X_T(k)}{Y_T(k)} \quad k \in 0, \dots, N-1 \quad (4.10)$$

Η ισοστάθμιση εκφράζεται τώρα ως:

$$\hat{X}_{RX}(k) = Y(k) \times He(k) \quad k \in 0, \dots, N-1 \quad (4.11)$$

Αυτό το σημείο διαφέρει ελάχιστα από τον αλγόριθμο LS. Ο λόγος για τον οποίο έγινε αυτή η τροποποίηση, είναι ότι η ισοστάθμιση γίνεται με έναν (μιγαδικό) πολλαπλασιασμό, ενώ στον αρχικό αλγόριθμο γίνεται με μία (μιγαδική) διαίρεση. Η εκτίμηση και στις δύο περιπτώσεις γίνεται με μία (μιγαδική) διαίρεση. Λαμβάνοντας υπόψη ότι η ισοστάθμιση επιτελείται στα περισσότερα σύμβολα, υπάρχει κέρδος στην ταχύτητα των υπολογισμών καθώς ο πολλαπλασιασμός είναι πιο απλή υπολογιστικά πράξη από τη διαίρεση.

Μία άλλη παρατήρηση είναι πως αν είχε εφαρμοστεί ο αρχικός αλγόριθμος, θα μπορούσε να είχε χρησιμοποιηθεί το ίδιο engine για την εκτίμηση και την ισοστάθμιση. Ωστόσο, πέραν της βραδύτερης πράξης της διαίρεσης που θα επιτελούταν κατά την ισοστάθμιση, ο διαχωρισμός των σταδίων estimation και compensation σε δύο ανεξάρτητες engines, αποτελεί υποδομή για μία μελλοντική υλοποίηση του *Decision Feedback Estimation*. Η τελευταία είναι μία τεχνική για την βελτίωση της εκτίμησης καναλιού, ιδιαίτερα σε μεταβαλλόμενα περιβάλλοντα.²⁷

4.9.1 Η Top-Level Όψη

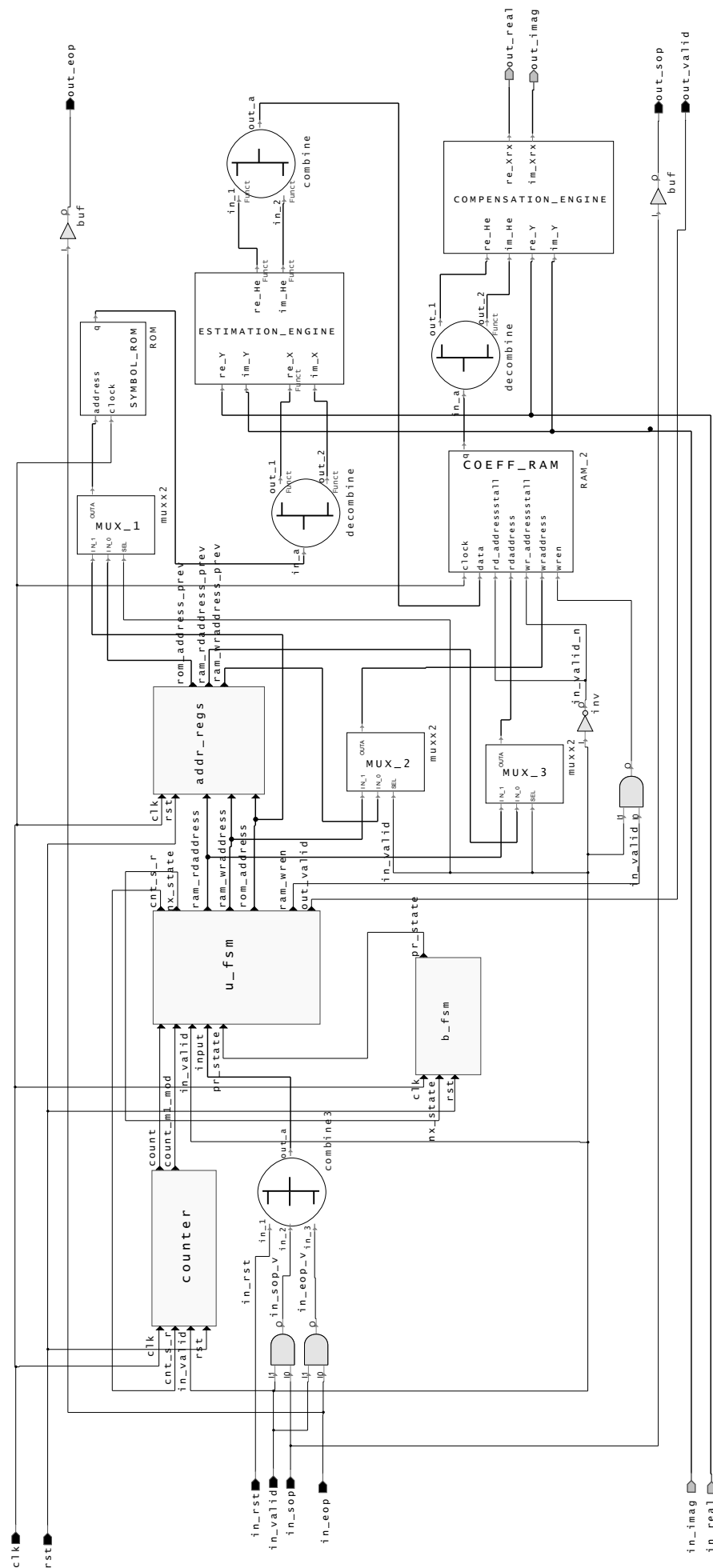
Βασικές οντότητες είναι: η μηχανή καταστάσεων (process fsm, b_fsm), η μνήμη ROM (SYMBOL_ROM), η μνήμη RAM -dual port (COEFF_RAM), το σύστημα μετρητών (process counter), ο υπολογιστής εκτίμησης (ESTIMATION_ENGINE), ο υπολογιστής ισοστάθμισης (COMPENSATION_ENGINE). Το top level διάγραμμα του κυκλώματος φαίνεται στο σχήμα 4.22.

Καθώς το module επιτελεί τόσο τη διαδικασία της εκτίμησης, όσο και αυτή της ισοστάθμισης καναλιού, υπάρχει η αναγκαιότητα συντονισμού των λειτουργιών και χρονοπρογραμματισμού των υποσυστημάτων του κυκλώματος. Αυτή είναι η κύρια εργασία της μηχανής κατάστασης. Τα estimation και compensation engines αποτελούν τον υπολογιστικό πυρήνα της εκτίμησης και της ισοστάθμισης καναλιού, αντίστοιχα. Η μνήμη ROM χρησιμοποιείται για την αποθήκευση του training block και διαβάζεται κατά τη διαδικασία της εκτίμησης. Η μνήμη RAM είναι δύο εισόδων, μία εγγραφής και μια ανάγνωσης. Χρησιμοποιείται τόσο κατά την εκτίμηση, όπου γράφονται οι εκτιμώμενοι συντελεστές του καναλιού, όσο και κατά την ισοστάθμιση, όπου διαβάζονται οι εκτιμώμενοι συντελεστές. Θα μπορούσε να είχε γίνει χρήση RAM μιας εισόδου καθώς στην παρούσα υλοποίηση δεν γράφεται και διαβάζεται ταυτόχρονα, ωστόσο η τρέχουσα αρχιτεκτονική είναι πιο απλή και αποτελεί υποδομή για μελλοντική υλοποίηση του *Decision Feedback Estimation*.

Η πρώτη διεργασία που επιτελείται χρονικά, είναι η εκτίμηση καναλιού. Το datapath της εισόδου (in_real, in_imag) και η έξοδος της ROM, στην οποία είναι αποθηκευμένο το training block, συνδέονται στο estimation engine. Σε κάθε κύκλο εξέρχεται από το engine, ένας συντελεστής του διανύσματος εκτίμησης $\vec{H}e$ στα σήματα realHe και imagHe. Οι συντελεστές αυτοί γράφονται στη RAM, μέσω της αντίστοιχης εισόδου.

Μετά τη λήψη και του τελευταίου subcarrier του training block, ακολουθεί η ισοστάθμιση. Το datapath της εισόδου συνδέεται στο compensation engine, καθώς επίσης και η έξοδος ανάγνωσης της RAM. Στην έξοδο του engine εξέρχονται ισοσταθμισμένα subsymbols έτοιμα προς αποδιαμόρφωση, τα οποία οδηγούνται στις θύρες εξόδου out_real, out_imag. Η διαδικασία της ισοστάθμισης επαναλαμβάνεται σε κάθε OFDM σύμβολο μέχρι το πέρας του frame, το οποίο σηματοδοτείται από το σήμα εισόδου in_rst.

²⁷ Στο *Decision Feedback Estimation*, οι συντελεστές του καναλιού, ανανεώνονται σε κάθε OFDM symbol παρέχοντας αυξημένη ανεκτικότητα σε σφάλματα εκτίμησης. Για να γίνει αυτό το εκάστοτε αποδιαμορφωμένο κατά QAM subcarrier, επαναδιαμορφώνεται και ανατροφοδοτείται στον channel estimator. Δηλαδή τα data symbols χρησιμοποιούνται σαν training και η διαδικασία της εκτίμησης είναι συνεχής και εκτελείται παράλληλα με αυτή της ισοστάθμισης.



Σχήμα 4.22: CEQ Top-Level διάγραμμα

Για το χειρισμό του μηχανισμού του frontpressure εισόδου, το σήμα `in_valid` καθορίζει τη λειτουργία του συστήματος μετρητών. Ακόμα, το αντίστροφο σήμα `in_valid_n`, τροφοδοτείται στην είσοδο `addressstall` της ROM και στις εισόδους `rd_addressstall` και `wr_addressstall` της RAM. Επίσης για τον ίδιο λόγο, υπάρχουν 3 βοηθητικοί latch (`process addr_regs`) που συγκρατούν την προηγούμενη διεύθυνση της ROM, της RAM ανάγνωσης και εγγραφής, καθώς και 3 MUX (`MUX_1`, `MUX_2`, `MUX_3`) που ελέγχονται από το `in_valid` και επιλέγουν είτε την τρέχουσα είτε την προηγούμενη τιμή των διευθύνσεων. Με αυτόν τον τρόπο το σύστημα ανταποκρίνεται σε όλες τις πιθανές περιπτώσεις άσκησης του μηχανισμού frontpressure από το upstream. Κατά την κανονική λειτουργία, το frontpressure της εισόδου ασκείται ανάμεσα στα OFDM σύμβολα λόγω της αφαίρεσης του cyclic prefix, αλλά και ανάμεσα στα OFDM frame λόγω της αφαίρεσης των training symbol του συγχρονισμού.

Για την άσκηση του frontpressure στα downstream modules υπεύθυνο είναι το σήμα `out_valid`, το οποίο ελέγχεται από το πάνω τμήμα της μηχανής κατάστασης. Το `out_valid` είναι '1' μόνο όταν το κύκλωμα βρίσκεται σε κατάσταση ισοστάθμισης και το `in_valid` είναι '1'. Σε κάθε άλλη περίπτωση είναι '0'. Κατά την κανονική λειτουργία, το frontpressure της εξόδου ασκείται τόσο στις περιπτώσεις που ασκείται το frontpressure της εισόδου, αλλά και κατά τη διαδικασία της εκτίμησης καναλιού, αφαιρώντας ουσιαστικά το training block από τα downstream δεδομένα.

Τα σήματα εισόδου `in_sop` και `in_eop` χρησιμοποιούνται για τον έλεγχο της μετάβασης των καταστάσεων. Ωστόσο λαμβάνονται τα `in_sop_v` και `in_eop_v`, περνώντας τα αντίστοιχα σήματα από μία πύλη AND με το σήμα `in_valid`. Αυτό είναι απόρροια του πρωτοκόλλου (Avalon ST) που έχει υλοποιηθεί σύμφωνα με το οποίο όλα τα σήματα εισόδου επικυρώνονται με το `in_valid`.

Τα σήματα `out_sop` και `out_eop` ακολουθούν τα αντίστοιχα σήματα εισόδου καθώς το I/O latency είναι 0. Ωστόσο η εγκυρότητά τους σηματοδοτείται από το σήμα `out_valid`.

Σύστημα μετρητών

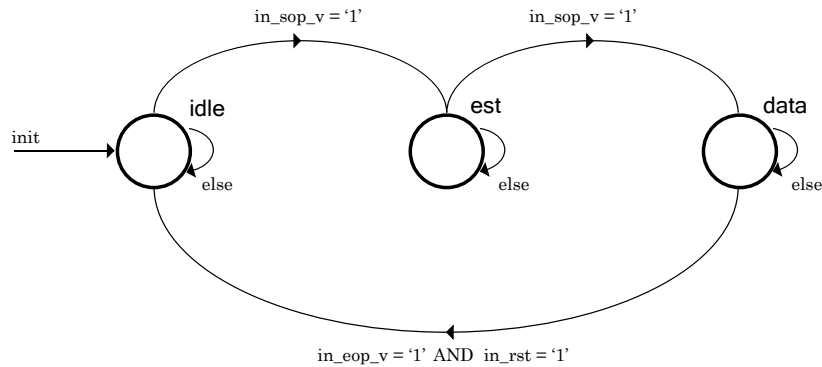
Το σύστημα μετρητών αποτελείται από έναν μετρητή ελεγχόμενο από τα σήματα `cnt_s_r` και `in_valid` και υλοποιείται στο `process counter`. Η έξοδος είναι ο κύριος μετρητής στο σήμα `count` και ένας βοηθητικός μετρητής `count_m1_mod` με το εύρος μέτρησής τους να είναι $[0, N_{SC} - 1]$. Η μέτρηση γίνεται με κυκλικό τρόπο, δηλαδή οι μετρητές επιστρέφουν αυτόματα στο 0 μετά το $N_{SC} - 1$. Ο δεύτερος μετρητής βρίσκεται κατά μία θέση πριν τον πρώτο, με κυκλικό τρόπο. Δηλαδή όταν ο `count = 0`, ο `count_m1_mod = N_{SC} - 1`. Οι μετρητές αυτοί χρησιμοποιούνται για την οδήγηση των διευθύνσεων των ROM και RAM. Η επιλογή τους γίνεται από το πάνω τμήμα της μηχανής κατάστασης.

Όταν το σήμα `cnt_s_r` είναι '1' οι μετρητές δουλεύουν κανονικά δεδομένου ότι το `in_valid` είναι επίσης '1'. Αν το `in_valid` είναι '0' τότε οι δύο μετρητές παγώνουν στην προηγούμενη τιμή τους. Όταν το `cnt_s_r` τεθεί στο '0', τότε ο `count` τίθεται στο 1 και κατ' επέκταση ο `count_m1_mod` στο 0. Οι τιμές αυτές μένουν σταθερές για όσο το `cnt_s_r` είναι κάτω. Συνεπώς το σήμα αυτό αποτελεί ένα συνδυασμό reset και enable.

Μηχανή καταστάσεων

Η μηχανή καταστάσεων είναι ο «εγκέφαλος» του module, συντονίζοντας τις υπομονάδες και οδηγώντας τα σήματα ελέγχου και εξόδου. Υπάρχουν 3 καταστάσεις: `idle`, `est`, `data`, ενώ η μηχανή είναι τύπου Mealy. Τα processes `u_fsm`, `b_fsm`, υλοποιούν το πάνω και το κάτω τμήμα της μηχανής, αντίστοιχα. Το διάγραμμα καταστάσεων απεικονίζεται στο Σχήμα 4.23.

Το κύκλωμα αρχίζει σε `idle` κατάσταση, όπου αναμένει για το πρώτο σύμβολο, το training block. Εδώ η είσοδος εγγραφής της RAM είναι απενεργοποιημένη, με το `wren` να είναι στο '0' και το σύστημα μετρητών είναι παγωμένο με το `cnt_s_r` να είναι στο '0'. Η διεύθυνση της ROM (`rom_address`) έχει τεθεί στο 0 με αποτέλεσμα να προ-διαβάζεται ο συντελεστής του πρώτου subsymbol του training block. Στο 0 έχουν τεθεί και οι διευθύνσεις `ram_rdaddress` και `ram_wraddress`. Το `out_valid` είναι κάτω, καθώς δεν εξέρχονται έγκυρα δεδομένα από το module. Μόλις σηκωθεί το `in_sop_v` τα σήματα `wren` και `cnt_s_r` τίθε-



Σχήμα 4.23: Διάγραμμα καταστάσεων (CEQ)

νται στο '1' ενεργοποιώντας την εγγραφή στη RAM και ξεκλειδώνοντας το σύστημα μετρητών, αντίστοιχα. Ταυτόχρονα το σήμα `rom_address` συνδέεται με το `count` και το `ram_waddress` με το `count_m1_mod`. Δηλαδή, η διεύθυνση εγγραφής στη RAM είναι κατά ένα μικρότερη από αυτή της ανάγνωσης από τη ROM, λόγω του ενός κύκλου καθυστέρησης για την ανάγνωση της ROM.

Στον επόμενο κύκλο η μηχανή μεταβαίνει στην κατάσταση `est`. Εδώ παραμένει καθ' όλη τη διάρκεια μετάδοσης του training block. Τα σήματα ελέγχου και διευθύνσεων παραμένουν ως είχαν στον τελευταίο κύκλο `idle`, όπου έγινε η αλλαγή κατάστασης. Οι εκτιμώμενοι συντελεστές όπως εξέρχονται από το estimation engine, αποθηκεύονται στη RAM. Όταν σηκωθεί το `in_sop_v` σηματοδοτείται η έναρξη μετάδοσης των data symbols και η μηχανή μεταβαίνει σε κατάσταση `data`. Ταυτόχρονα η είσοδος εγγραφής της RAM απενεργοποιείται θέτοντας το `wren` στο '0', ενώ το σήμα `ram_raddress` οδηγείται από το `count`. Επιπλέον το `out_valid` συνδέεται με το `in_valid` σηματοδοτώντας την εγκυρότητα των δεδομένων εξόδου.

Στην κατάσταση `data` γίνεται η ισοστάθμιση των subsymbols. Τα σήματα ελέγχου και οι διευθύνσεις παραμένουν ως είχαν κατά τη μετάβαση στην τρέχουσα κατάσταση εκτός από το `ram_waddress` όπου τίθεται στο 0. Καθώς η διεύθυνση ανάγνωσης της RAM οδηγείται από τον `count`, στην έξοδό της εξέρχονται με τη σειρά, οι συντελεστές του καναλιού. Οι τελευταίοι τροφοδοτούνται μαζί με τα δεδομένα εισόδου στο compensation engine, στην έξοδό του εξέρχονται τα αντισταθμισμένα subsymbols. Ο `count` αναδιπλώνεται στο 0 μετά το $N_{SC} - 1$ subsymbol, με αποτέλεσμα στο επόμενο data symbol να τροφοδοτούνται οι ίδιοι συντελεστές στο compensation engine και να ισοσταθμίζονται ακολούθως τα δεδομένα εισόδου. Η μηχανή παραμένει σε αυτή την κατάσταση μέχρι να τεθούν στο '1' τα σήματα `in_eop_v` και `in_rst`. Εκείνη τη στιγμή σηματοδοτείται το τέλος του frame (τελευταίο subsymbol του τελευταίου συμβόλου του frame). Το σήμα `cnt_s_r` τίθεται στο '0' μηδενίζοντας²⁸ και απενεργοποιώντας το σύστημα μετρητών. Επιπλέον, η διεύθυνση της ROM `rom_address` τίθεται στο 0 προ-διαβάζοντας τον συντελεστή του πρώτου subsymbol του training block. Το κύκλωμα έχει προετοιμαστεί για τον επόμενο κύκλο εκτίμησης. Εδώ γίνεται μετάβαση στην κατάσταση `idle` και ο κύκλος επαναλαμβάνεται.

Τα Επιμέρους Υποσυστήματα

4.9.2 Μνήμη Συντελεστών [CEQ.COEFF_RAM]

```

1 component ch_coef_ram is
2   port(
3     clock : in std_logic := '1';
4     data : in std_logic_vector(31 downto 0);
5     raddress : in std_logic_vector(9 downto 0);
  
```

²⁸Όπως αναφέρθηκε ο `count` τίθεται στο 1 και παγώνει εκεί.

```

6   rd_addressstall : in std_logic := '0';
7   wr_addressstall : in std_logic := '0';
8   wraddress      : in std_logic_vector(9 downto 0);
9   wren          : in std_logic := '0';
10  q              : out std_logic_vector(31 downto 0)
11  );
12 end component;
```

Τύπος	Altsyncram • Dual Port
Εύρος θύρας διεύθυνσης	10 bit
Εύρος θύρας εισόδου/εξόδου	32 bit

Πίνακας 4.16: Τεχνικά Χαρακτηριστικά Altera RAM (CEQ)

Η RAM είναι «τυποποιημένη» Altera RAM δύο θυρών, μίας εισόδου και μιας εξόδου. Η RAM διαθέτει επιπλέον, θύρα ενεργοποίησης εγγραφής *wren*, καθώς και θύρες παγώματος της εγγραφής *wr_addressstall* και της ανάγνωσης *rd_addressstall*. Τα χαρακτηριστικά της συνοψίζονται στον πίνακα 4.16.

4.9.3 Μνήμη Συμβόλων [CEQ.SYMBOL_ROM]

```

1 component symbol_rom is
2   port(
3     address : in std_logic_vector(9 downto 0);
4     addressstall_a : in std_logic := '0';
5     clock : in std_logic := '1';
6     q : out std_logic_vector(31 downto 0)
7   );
8 end component;
```

Τύπος	Altsyncrom • Single Port
Εύρος θύρας διεύθυνσης	10 bit
Εύρος θύρας εισόδου/εξόδου	32 bit

Πίνακας 4.17: Τεχνικά Χαρακτηριστικά Altera ROM (CEQ)

Η ROM είναι «τυποποιημένη» Altera ROM μίας θύρας εξόδου. Η ROM διαθέτει επιπλέον, θύρα παγώματος της ανάγνωσης *addressstall_a*. Τα χαρακτηριστικά της συνοψίζονται στον πίνακα 4.17.

4.9.4 Μηχανή Εκτίμησης [CEQ.ESTIMATION_ENGINE]

Το υποσύστημα αυτό είναι ένας υπολογιστής της αντίστροφης εκτίμησης καναλιού. Υπολογίζονται τα μεγέθη re_{He_k} και im_{He_k} , τα οποία εκφράζουν το πραγματικό και φανταστικό μέρος του $He(k)$. Σε κάθε κύκλο εισέρχονται στην είσοδο ένα ληφθέν (training) subsymbol, Y_{T_k} και ένα αποθηκευμένο training subsymbol, X_{T_k} . Στην έξοδο υπολογίζεται ένας συντελεστής του καναλιού He_k . Οι αναλυτικοί τύποι υπολογισμού τους, όπως προκύπτουν από την (4.10) είναι:

$$re_{He_k} = \frac{re_{X_{T_k}} \times re_{Y_{T_k}} + im_{X_{T_k}} \times im_{Y_{T_k}}}{De_k}$$

$$im_{He_k} = \frac{im_{X_{T_k}} \times re_{Y_{T_k}} - re_{X_{T_k}} \times im_{Y_{T_k}}}{De_k}$$

$$De_k = re_{Y_{T_k}} \times re_{Y_{T_k}} + im_{Y_{T_k}} \times im_{Y_{T_k}}$$

Η πολυπλοκότητα $\forall k$ είναι :

$$6 \times \text{MULT}, 4 \times \text{ADD}, 2 \times \text{DIV}$$

Το datapath του κυκλώματος μαζί με τις ενδιάμεσες ακρίβειες απεικονίζονται στο Σχήμα 4.24. Οι αντιστοιχίες μεταξύ των μεγεθών των παραπάνω σχέσεων και των σημάτων του σχήματος, έχουν ως εξής:

$$\begin{array}{ll} \text{re_}X_{T_k} \rightarrow \text{re_}X & \text{im_}X_{T_k} \rightarrow \text{im_}X \\ \text{re_}Y_{T_k} \rightarrow \text{re_}Y & \text{im_}Y_{T_k} \rightarrow \text{im_}Y \\ \text{re_}He_k \rightarrow \text{re_}He & \text{im_}He_k \rightarrow \text{im_}He \end{array}$$

Αριθμητική

Οι είσοδοι $\text{re_}Y_{T_k}$ και $\text{im_}Y_{T_k}$ είναι τύπου $\langle s | 16 \rangle$. Ωστόσο για τον υπολογισμό της εκτίμησης He_k πρέπει να γίνει χρήση fixed point αριθμητικής. Όπως φαίνεται και στο datapath, ο αριθμητής της διαίρεσης μετατρέπεται σε $\langle s | 42 . 11 \rangle$ προτού εκτελεστεί η πράξη. Δηλαδή έχει επιλεγεί δεκαδικό μέρος των 11 bit. Εν τέλει ο τύπος εξόδου του estimation engine, για τα $\text{re_}He_k$ και $\text{im_}He_k$, είναι $\langle s | 16 . 11 \rangle$. Η μετατροπή στον τελικό τύπο γίνεται με κορεσμό, ώστε τιμές που δεν «χωράνε» στο δυναμικό εύρος του $\langle s | 16 . 11 \rangle$ να στρογγυλοποιούνται στην πλησιέστερη. Ο τύπος αυτός κάνει δυνατή την αποθήκευση ενός μιγαδικού συντελεστή He_k σε λέξη των 32 bit, εξοικονομώντας χώρο στη μνήμη αποθήκευσης των συντελεστών. Ακόμα πρέπει να σημειωθεί πως η Altera Ram, που χρησιμοποιείται, δεν παρέχει απεριόριστη ελευθερία όσων αφορά το εύρος της λέξης. Εν προκειμένω, η επόμενη δυνατή επιλογή εύρους λέξης μετά τα 32 bit, είναι τα 64 bit.

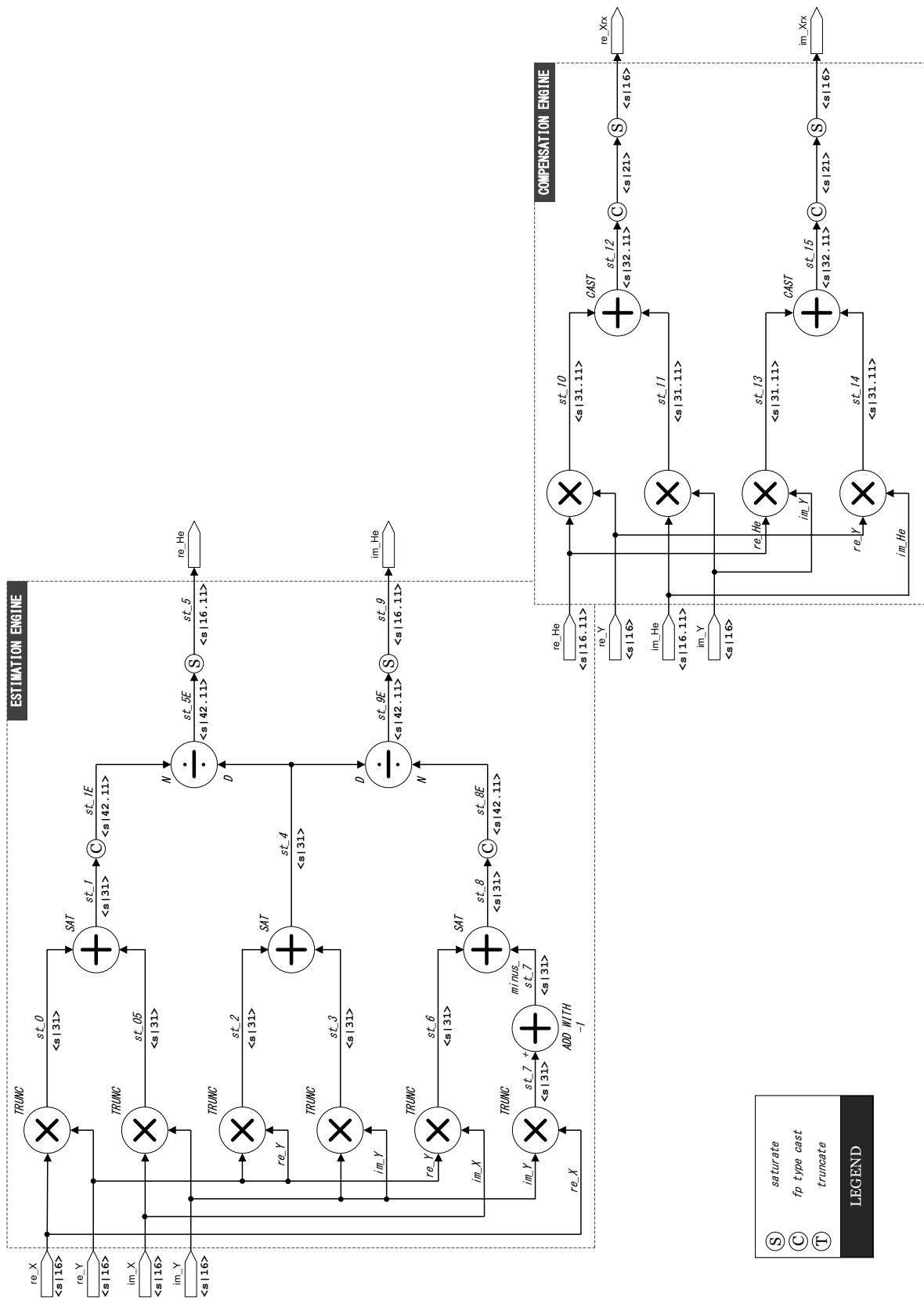
Για την πιστοποίηση της επάρκειας του συγκεκριμένου τύπου αποθήκευσης για τους συντελεστές $\text{re_}He_k$ και $\text{im_}He_k$, πραγματοποιήθηκε ανάλυση σφάλματος κάνοντας χρήση μοντέλου αριθμητικής του κυκλώματος στο MATLAB. Με τη βοήθεια του *Fixed Point Toolbox*, οι πράξεις στο μοντέλο πραγματοποιούνται ακριβώς όπως στο κύκλωμα. Με αυτόν τον τρόπο διευκολύνεται η αποτίμηση ακρίβειας του κυκλώματος, καθώς η ανάλυση γίνεται στο MATLAB. Οι εκάστοτε αλλαγές πραγματοποιούνται άμεσα, επιταχύνοντας τη διαδικασία ανάπτυξης. Εφόσον επιτευχθεί η επιθυμητή ακρίβεια στο μοντέλο, οι πράξεις μεταγράφονται σε γλώσσα VHDL, ακολουθώντας τα ίδια βήματα.

Στο MATLAB παρήχθησαν ορισμένα ενδεικτικά κανάλια multipath, χωρίς και με AWGN. Στη συνέχεια πραγματοποιήθηκε εκτίμηση του καναλιού με :

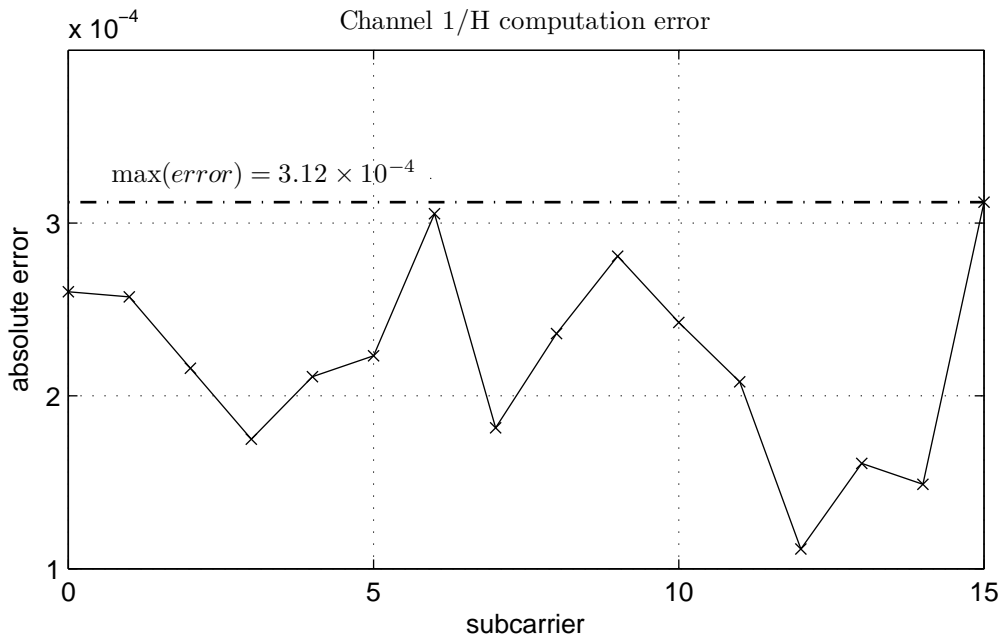
- α) Το μοντέλο και
- β) θεωρητικό υπολογισμό ακρίβειας double

Ακρίβεια Εξόδου Κατόπιν υπολογίστηκε το σφάλμα μεταξύ θεωρητικού υπολογισμού και μοντέλου. Το εύρος του δεκαδικού μέρους αφέθηκε ως παράμετρος. Μετά από διάφορες δοκιμές αποφασίστηκε πως η τιμή των 11 bit είναι κατάλληλη. Στο Σχήμα 4.25 απεικονίζεται το σφάλμα υπολογισμού (μεταξύ μοντέλου και θεωρητικής εκτίμησης) για ένα τυπικό κανάλι παρουσία AWGN, για ένα σύστημα με 16 subcarrier. Το μέγιστο σφάλμα υπολογισμού (ακρίβεια εξόδου) εν γένει είναι μικρότερο του 4×10^{-4} , τιμή που κρίνεται επαρκής.

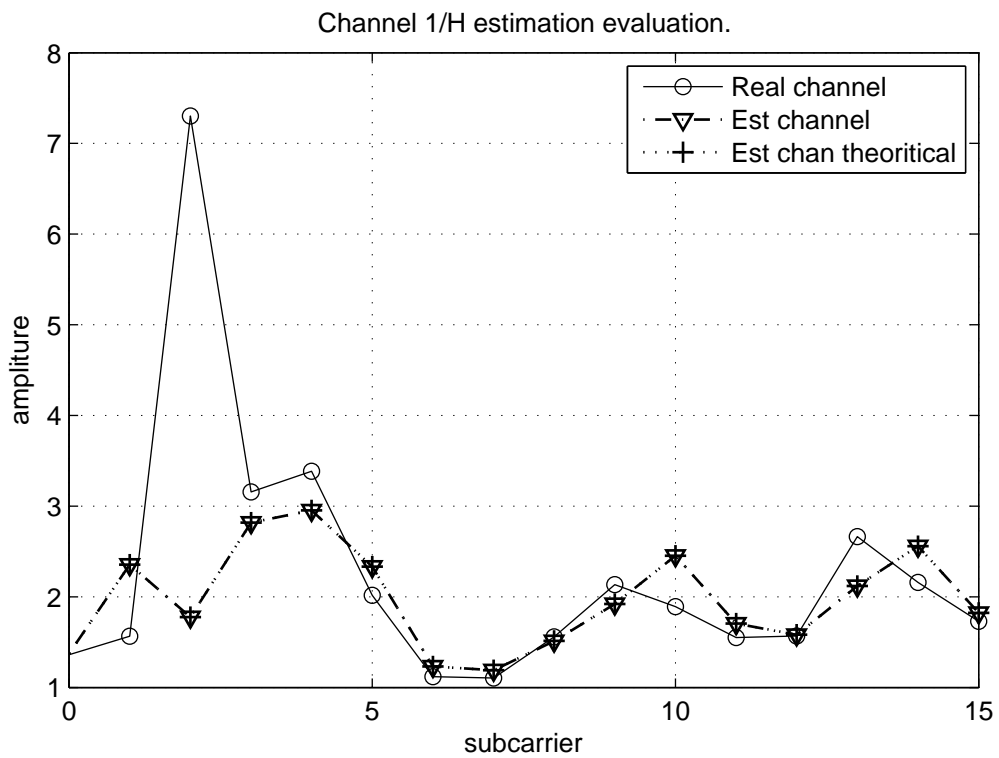
Παρατηρήθηκε ότι η παρουσία AWGN δεν επηρεάζει την ακρίβεια των υπολογισμών. Ωστόσο, όπως είναι αναμενόμενο, σε αυτή την περίπτωση και οι δύο εκτιμήσεις αποκλίνουν από την πραγματική αντίστροφη συνάρτηση μεταφοράς του καναλιού. Το σφάλμα εκτίμησης λόγω του αλγορίθμου LS είναι αρκετά μεγαλύτερο από το σφάλμα υπολογισμού, όπως μπορεί να φανεί στο Σχήμα 4.26, όπου συγκρίνονται οι δύο εκτιμήσεις με την πραγματική αντίστροφη συνάρτηση μεταφοράς.



Σχήμα 4.24: Μηχανές υπολογισμού του CEQ



Σχήμα 4.25: Σφάλμα υπολογισμών κατά την εκτίμηση ενός τυπικού καναλιού



Σχήμα 4.26: Σύγκριση εκτίμησης καναλιού μεταξύ μοντέλου και θεωρίας, παρουσία AWGN

Η χρήση του τύπου $\langle s | 16 \cdot 11 \rangle$ έχει και μία άλλη επίπτωση, ως προς το μέγιστο μέτρο ενός συντελεστή που μπορεί να αποθηκευτεί. Καθώς διατίθενται 4 bit για το ακέραιο μέρος (εξαιρώντας το bit προσήμου) το απόλυτο δυναμικό εύρος είναι $|re_{He_k}|, |im_{He_k}| \in [0, 15]$. Στη χειρότερη περίπτωση όπου το ένα στέλεχος του μιγαδικού αριθμού που αναπαριστά την αντίστροφη συνάρτηση μεταφοράς He_k , είναι κοντά στο μηδέν και το άλλο στο μέγιστο (15), τότε το μέγιστο μέτρο της αντίστροφης συνάρτησης θα είναι $\max(|He_k|) = 15$. Για το μέτρο της συνάρτησης μεταφοράς θα ισχύει $\min(|H_k|) = 0.0667$. Αυτή είναι η ελάχιστη τιμή, ώστε να μη συμβαίνει ψαλιδισμός της $|He_k|$. Ωστόσο σε τόσο μεγάλες εξασθενίσεις είναι, ούτως ή άλλως, πρακτικά αδύνατη η σωστή εκτίμηση καναλιού λόγω ύπαρξης θορύβου. Γι' αυτό, το δυναμικό εύρος που έχει επιλεγθεί κρίνεται επαρκές για τη συγκεκριμένη εφαρμογή.

4.9.5 Μηχανή Ισοστάθμισης [CEQ.COMPENSATION_ENGINE]

Σε αυτό το υποσύστημα του CEQ, γίνεται η ισοστάθμιση των ληφθέντων data symbol. Σε κάθε κύκλο τροφοδοτείται στην είσοδο ένα ληφθέν subsymbol Y_k και ο αντίστοιχος συντελεστής καναλιού He_k . Στην έξοδο παρέχεται το αντισταθμισμένο subsymbol X_{RXk} . Οι αναλυτικοί τύπου υπολογισμού, όπως προκύπτουν από την (4.11), είναι:

$$\begin{aligned} re_{X_{RXk}} &= re_{He_k} \times re_{Y_k} - im_{He_k} \times im_{Y_k} \\ im_{X_{RXk}} &= re_{He_k} \times im_{Y_k} + im_{He_k} \times re_{Y_k} \end{aligned}$$

Η πολυπλοκότητα $\forall k$ είναι:

$$4xMULT, 2xADD$$

Στο Σχήμα 4.24 απεικονίζεται το datapath του κυκλώματος, μαζί με τις ενδιάμεσες ακρίβειες εργασίας. Η αντιστοιχία των σημάτων με τους παραπάνω τύπους έχει ως εξής:

$$\begin{array}{ll} re_{X_{RXk}} \rightarrow re_{Xrx} & im_{X_{RXk}} \rightarrow im_{Xrx} \\ re_{Y_k} \rightarrow re_Y & im_{Y_k} \rightarrow im_Y \\ re_{He_k} \rightarrow re_{He} & im_{He_k} \rightarrow im_{He} \end{array}$$

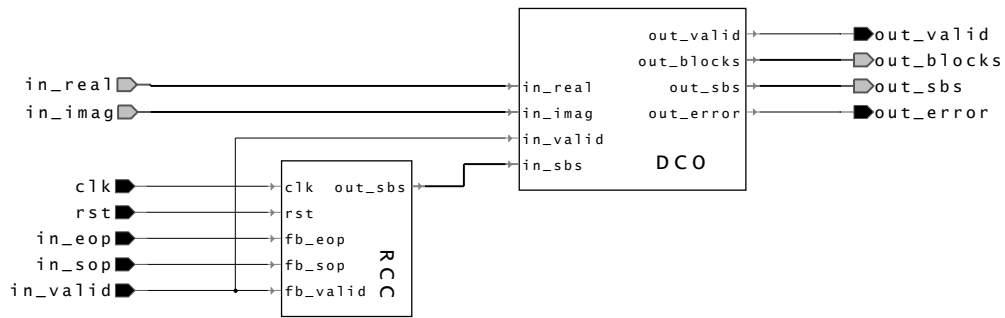
Όσον αφορά στην εκτέλεση των πράξεων, εδώ λαμβάνουν χώρα πολλαπλασιασμοί και προσθέσεις. Η ακρίβεια κρατείται στο μέγιστο δυνατό (τα 11 δεκαδικά bit) μέχρι και την έξοδο του adder. Καθώς τα samples εξόδου θα πρέπει να μετατραπούν σε $\langle s | 16 \rangle$, γίνεται πρώτα αποκοπή των δεκαδικών ψηφίων οδηγώντας σε τύπο $\langle s | 21 \rangle$ και τέλος μετατροπή σε $\langle s | 16 \rangle$ με κορεσμό.

4.10 Symbol Demapper [DMP]

Σε αυτό το module γίνεται η αποδιαμόρφωση κατά QAM. Στην είσοδο `in_real` και `in_imag` εισέρχεται σε κάθε clock, ένα μιγαδικό subsymbol. Στην έξοδο `out_blocks` εξέρχεται το k-block που αντιστοιχεί στο αποδιαμορφωθέν subsymbol. Ταυτόχρονα στην έξοδο `out_sbs` παρέχεται το ωφέλιμο εύρος του k-block, δηλαδή ο αριθμός k. Οι πιθανές τιμές είναι $k \in \{2, 4, 6\}$.

Βασικό στοιχείο για την πραγματοποίηση της αποδιαμόρφωσης είναι η πληροφορία του μεγέθους του αστερισμού σύμφωνα με το οποίο έχει διαμορφωθεί το κάθε subcarrier. Όπως έχει αναφερθεί και στο Κεφάλαιο 3, το κάθε subcarrier μπορεί να έχει διαφορετικό τύπο διαμόρφωσης. Οι πιθανοί τύποι διαμόρφωσης είναι QAM-4, QAM-16, QAM-64. Η πληροφορία αυτή είναι κωδικοποιημένη στο Modulation Pattern, το οποίο είναι γνωστό στον DMP.

4.10.1 Η Top-Level Όψη



Σχήμα 4.27: DMP Top-Level διάγραμμα

Όπως φαίνεται και στο Σχήμα 4.27 που απεικονίζει το toplevel κύκλωμα του DMP, υπάρχουν δύο βασικά υποσυστήματα. Ο RCC (Rx Constellation Controller), που είναι υπεύθυνος για την παραγωγή του k (στο σήμα `out_sbs`) και ο DCO (Deconstallator), που επιτελεί την καθ' αυτό αποδιαμόρφωση. Τα σήματα `in_sop`, `in_eop` και `in_valid` του Avalon χρησιμοποιούνται από τον RCC για το συγχρονισμό του modulation pattern με τα εισερχόμενα subsymbols. Τα τελευταία τροφοδοτούνται στον DCO, ο οποίος παράγει τις εξόδους του κυκλώματος.

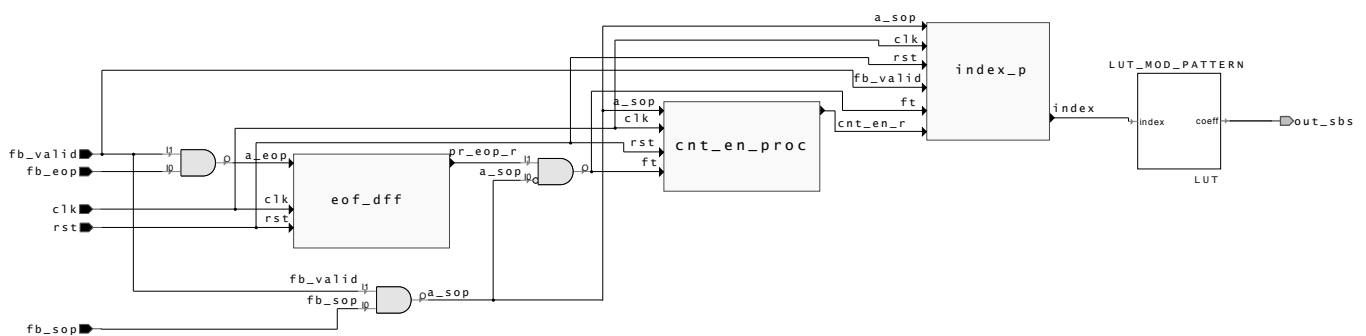
Τα Επιμέρους Υποσυστήματα

4.10.2 RX Constellation Controller [DMP.RCC]

```

1 component rx_const_control is
2   port (
3     clk : in std_logic;
4     rst : in std_logic;
5     fb_eop : in std_logic;
6     fb_sop : in std_logic;
7     fb_valid : in std_logic;
8     out_sbs : out std_logic_vector (2 downto 0) -- generated symblocksize (2, 4, 6)
9   );
10 end component rx_const_control;

```



Σχήμα 4.28: DMP.RCC Top-Level διάγραμμα

Οι βασικές οντότητες εδώ είναι: η LUT με το modulation pattern (LUT_MOD_PATTERN), ο μετρητής διεύθυνσης (process index_p) και λογική για τον έλεγχο του μετρητή (processes cnt_en_proc, eop_dff). Στο Σχήμα 4.28 απεικονίζεται το top level κύκλωμα του RCC.

Το modulation pattern είναι αποθηκευμένο στη LUT. Η έξοδός της συνδέεται απευθείας στην έξοδο out_sbs. Η LUT υλοποιείται ως μία δομή array με σταθερές τιμές. Δεν υπάρχει ανάγκη για υλοποίηση πραγματικής ROM καθώς το μέγεθος είναι αρκετά μικρό (N_{SC} το πλήθος ακέραιοι). Η διεύθυνση της LUT παράγεται από την έξοδο index του counter.

Ο μετρητής (index_p) ελέγχεται από διάφορα σήματα ώστε να συγχρονίζεται με τα όρια του συμβόλου, όπως αυτό οριοθετείται με τα σήματα fb_sop, fb_eop. Η ενεργοποίηση του counter γίνεται όταν ισχύει (cnt_en_r = '1' AND ft = '0' AND fb_valid = '1'), διαφορετικά είναι παγωμένος στην προηγούμενη τιμή. Ο «μηδενισμός» του γίνεται σύγχρονα, όταν ισχύει a_eop = '1', οπότε ο μετρητής τίθεται στο 1. Η τιμή αυτή προκύπτει λόγω της καθυστέρησης ενός κύκλου από τη στιγμή που σηκώνεται το a_sop μέχρι να ενημερωθεί η τιμή του index. Ο counter είναι κυκλικός, δηλαδή επανεκκινεί από το 0, μετά την τιμή PATTERN_SIZE-1. Το PATTERN_SIZE είναι μια σταθερά που ισούται με το μέγεθος του modulation pattern και η τιμή της ορίζεται μέσω του μηχανισμού *Parameter Dispatching*.

Τώρα, όσον αφορά τα σήματα ελέγχου του counter: Το fb_valid χειρίζεται το frontpressure. Όταν δεν υπάρχει έγκυρο δεδομένο στην είσοδο η μέτρηση σταματά. Το cnt_en_r είναι ένα σήμα απενεργοποίησης της μέτρησης όταν δεν υπάρχει επόμενο σύμβολο, ενώ το σήμα fp είναι ένα βοηθητικό σήμα για την ίδια περίπτωση. Συγκεκριμένα το fp σηκώνεται όταν μετά από ένα έγκυρο eop δεν ακολουθεί έγκυρο sop. Σε αυτή την περίπτωση η μέτρηση σταματά άμεσα. Επιπλέον μέσω του cnt_en_proc το cnt_en_r τίθεται στο '0', απενεργοποιώντας μόνιμα τον counter. Στη συνέχεια, όταν συμβεί a_sop = '1', το cnt_en_r τίθεται στο '1' και ο counter ενεργοποιείται και πάλι.

4.10.3 Deconstellator [DMP.DCO]

```

1 component deconstella is
2   generic(
3     DIN_WIDTH_G : natural := 16
4   );
5   port (
6     --in
7     clk : in std_logic;
8     rst : in std_logic;
9     in_real : in std_logic_vector(DIN_WIDTH_G -1 downto 0);
10    in_imag : in std_logic_vector(DIN_WIDTH_G -1 downto 0);
11    in_valid : in std_logic; --apto top
12    in_sbs : in std_logic_vector(2 downto 0);
13    --out
14    out_valid : out std_logic;
15    out_blocks : out std_logic_vector(5 downto 0);
16    out_sbs : out std_logic_vector(2 downto 0);
17    -- out_error : invalid n_sblock/constellation type
18    out_error : out std_logic
19  );
20 end component;
```

Σε αυτό το υποσύστημα γίνεται η αποδιαμόρφωση κατά QAM. Στις εισόδους in_real, in_imag εισέρχεται ένα subsymbol σε κάθε clock. Ταυτόχρονα στην είσοδο in_sbs εισέρχεται το μέγεθος k που χαρακτηρίζει τον αστερισμό, σύμφωνα με τον οποίο έχει διαμορφωθεί το subsymbol. Στην έξοδο out_blocks παρέχεται το k-block, δηλαδή μία ακολουθία k bits που αντιστοιχεί στο subsymbol που ανιχνεύθηκε. Το εύρος του διαδρόμου είναι 6 bit, ώστε να καλύπτεται η πιο απαιτητική περίπτωση, αυτή του QAM-64. Στις περιπτώσεις όπου η διαμόρφωση είναι QAM-16 ή QAM-4, τα 2 ή τα 4 MSB του διαδρόμου είναι αδιάφορα

ή πλεονάζοντα. Τα υπόλοιπα bit, που φέρουν πληροφορία, λέγονται ωφέλιμα. Σε κάθε περίπτωση στην έξοδο `out_sbs` παρέχεται το ωφέλιμο εύρος του διαδρόμου, δηλαδή ο αριθμός k .

Αλγόριθμος

Για την αποδιαμόρφωση κατά QAM υλοποιήθηκε ο αλγόριθμος *Μεγίστης Πιθανοφάνειας (Maximum Likelihood, ML)*. Για κάθε ληφθέν subsymbol r , αναζητείται το σύμβολο του αστερισμού s_m που βρίσκεται πλησιέστερα, με την έννοια της ευκλείδειας απόστασης, σε αυτό. Αποδεικνύεται ότι αυτός είναι ο βέλτιστος φωρατής σε κανάλια AWGN, όταν η πιθανότητα μετάδοσης όλων των συμβόλων είναι η ίδια.²⁹

Για την εύρεση του συμβόλου που μεταδόθηκε, χρησιμοποιήθηκε ένας αλγόριθμος της οικογένειας *διαίρει-και-βασίλευε*. Το μιγαδικό επίπεδο του αστερισμού χωρίζεται σε 4 τεταρτημόρια. Με δύο συγκρίσεις είναι δυνατό να παρθεί απόφαση για το τεταρτημόριο στο οποίο βρίσκεται το r (έστω α -τεταρτημόριο). Αν ο τύπος διαμόρφωσης είναι QAM-4, το πρόβλημα έχει λυθεί, καθώς δεδομένης της τετραγωνικής μορφής των αστερισμών, κάθε σύμβολο βρίσκεται στο κέντρο ενός ξεχωριστού τεταρτημορίου.³⁰ Για άλλου τύπου διαμορφώσεις η διαδικασία επαναλαμβάνεται με παρόμοιο τρόπο: Το τεταρτημόριο του πρώτου σταδίου χωρίζεται εκ νέου σε 4 τεταρτημόρια (έστω β -τεταρτημόρια) και γίνεται αναζήτηση. Με δύο συγκρίσεις είναι δυνατό να βρεθεί το β -τεταρτημόριο στο οποίο ανήκει το r . Αν ο τύπος διαμόρφωσης είναι QAM-16, το πρόβλημα έχει λυθεί. Διαφορετικά, αν ο τύπος διαμόρφωσης είναι QAM-64, η διαδικασία επαναλαμβάνεται ακόμη μία φορά με παρόμοιο τρόπο και αποφασίζεται το γ -τεταρτημόριο, άρα και το ληφθέν σύμβολο.

Αρχιτεκτονική

Ακολουθώντας τα βήματα που περιγράφηκαν προηγουμένως, το κύκλωμα αποτελείται από 3 στάδια συγκρίσεων και ένα στάδιο ανακατασκευής. Σε κάθε στάδιο σύγκρισης λαμβάνεται η απόφαση για το (α, β, γ)-τεταρτημόριο, δεδομένου του αποτελέσματος του προηγούμενου σταδίου. Το στάδιο της ανακατασκευής περιλαμβάνει δομές LUT για την αντιστοίχιση της εσωτερικής κωδικοποίησης με την κωδικοποίηση Gray, σύμφωνα με την οποία αριθμούνται τα σύμβολα πάνω στον αστερισμό. Πριν την έξοδο υπάρχει ένας MUX, που επιλέγει τον κλάδο του `datapath` που θα συνδεθεί στην έξοδο, ανάλογα με την τιμή του `in_sbs`, δηλαδή ανάλογα με τον τύπο διαμόρφωσης. Ένα συνδυαστικό κύκλωμα παράγει το σήμα `out_error` όταν το `in_sbs` δεν περιέχει κάποια έγκυρη τιμή.

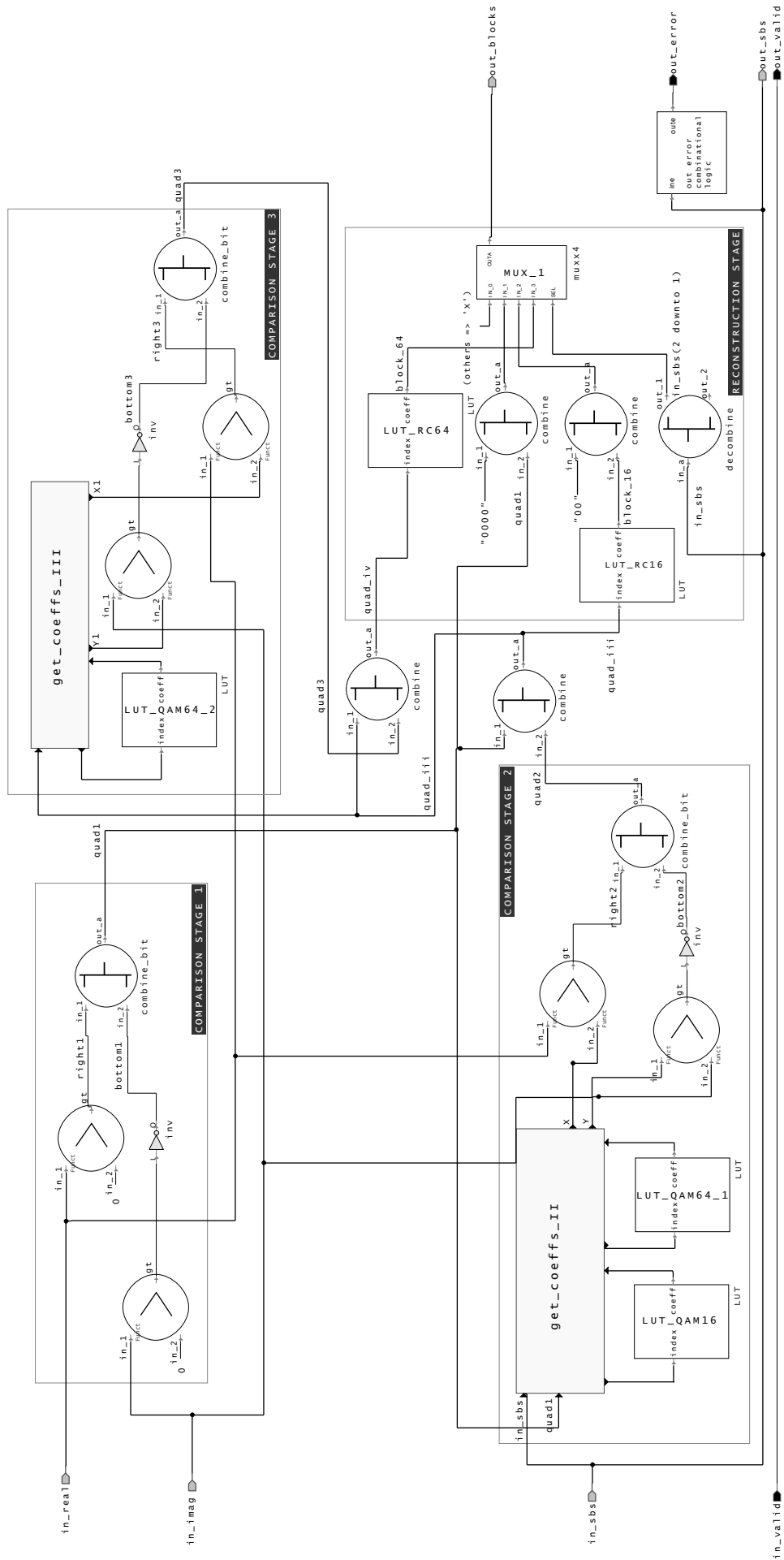
Η λειτουργία του κυκλώματος είναι ασύγχρονη, έχοντας έτσι I/O latency = 0. Το top-level διάγραμμα του κυκλώματος απεικονίζεται στο Σχήμα 4.29.

Στάδια σύγκρισης

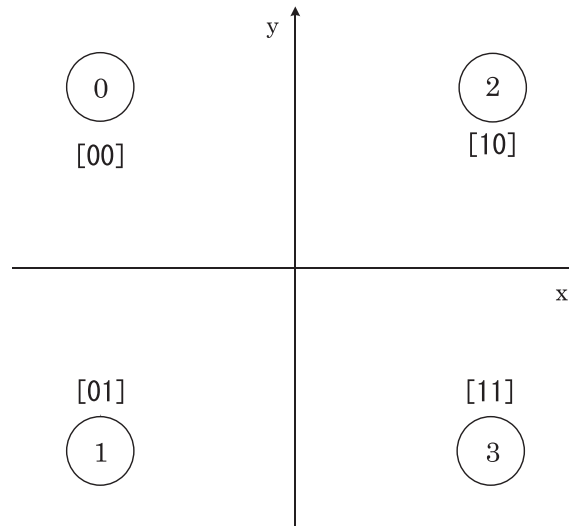
Σε κάθε στάδιο σύγκρισης υπάρχουν 2 συγκριτές των 16 bit, όπου ο καθένας έχει έξοδο ενός bit. Τα δύο αυτά bit συνενώνονται για να αποτελέσουν μία λέξη των 2 bit που κωδικοποιεί το τεταρτημόριο. Η κωδικοποίηση των τεταρτημορίων για κάθε στάδιο απεικονίζεται στο σχήμα 4.30, μέσα στις αγκύλες. Το πρώτο στάδιο είναι κοινό για όλους τους τύπους διαμορφώσεων και έχει ως αποτέλεσμα το σήμα `quad1`. Το δεύτερο χρησιμοποιείται από τις QAM-16, QAM-64 με αποτέλεσμα στο σήμα `quad2` και το τρίτο χρησιμοποιείται από την QAM-64 με αποτέλεσμα στο σήμα `quad3`. Το σήμα `quad_iii` είναι ίσο με `quad1 & quad2` (όπου με `&` συμβολίζεται ο τελεστής ένωσης), ενώ το σήμα `quad_iv` ισούται με `quad_iii & quad3`. Εν τέλει ως έξοδοι των σταδίων σύγκρισης νοούνται οι λέξεις `quad1`, `quad_iii`, `quad_iv` με εύρη 2, 4, 6 bit αντίστοιχα. Κάθε μία κωδικοποιεί όλα τα δυνατά σύμβολα για τις διαμορφώσεις QAM-4, QAM-16, QAM-64, αντίστοιχα.

²⁹Βλ. [PS02, § 7.5.3]

³⁰Περισσότερα για τους τετραγωνικούς αστερισμούς έχουν αναφερθεί στην § 2.2.1



Σχήμα 4.29: DMP.DCO Top-Level διάγραμμα



Σχήμα 4.30: Κωδικοποίηση τεταρτημορίων για τα στάδια σύγκρισης (DMP.DCO)

Στο πρώτο στάδιο το σήμα εισόδου συγκρίνεται με το σημείο της αρχής των αξόνων με συντεταγμένες (0,0). Ωστόσο στο δεύτερο στάδιο οι συγκρίσεις γίνονται με βάση το κέντρο του α-τεταρτημορίου που αποφασίστηκε στο πρώτο στάδιο. Οι τιμές των κέντρων είναι αποθηκευμένες σε δομές LUT (LUT_QAM16, LUT_QAM64_1). Η διαδικασία `get_coefs_II` δέχεται ως είσοδο το `in_sbs` και το `quad1` και επιστρέφει τις συντεταγμένες του κέντρου του τρέχοντος α-τεταρτημορίου (X , T) που αντιστοιχεί στον τύπο διαμόρφωσης της εισόδου. Κατά τον ίδιο τρόπο, στο τρίτο στάδιο οι συγκρίσεις γίνονται με βάση το κέντρο του β-τεταρτημορίου που αποφασίστηκε στο δεύτερο στάδιο. Μία LUT (LUT_QAM64_2) αποθηκεύει τους συντελεστές. Η διαδικασία `get_coefs_III` δέχεται ως είσοδο το `quad_iii` και επιστρέφει τους τις συντεταγμένες του κέντρου του τρέχοντος β-τεταρτημορίου (X_1 , T_1). Τα περιεχόμενα των παραπάνω LUT, παράγονται από το μηχανισμό των *Asset Dispatching* καθώς εξαρτώνται από τους αστερισμούς που χρησιμοποιούνται κατά τη μετάδοση.

Στάδιο ανακατασκευής

Περνώντας στο στάδιο της ανακατασκευής, υπάρχουν δύο LUT (LUT_RC16, LUT_RC64) που μεταφράζουν την εσωτερική κωδικοποίηση των συμβόλων σε κωδικοποίηση κατά Gray. Η LUT_RC16 μεταφράζει το `quad_iii` σε `block` διαμόρφωσης QAM-16 (`block_16`), ενώ η LUT_RC64 μεταφράζει το `quad_iv` σε `block` διαμόρφωσης QAM-64 (`block_64`). Στην περίπτωση του QAM-4 δε χρειάζεται μετάφραση καθώς η εσωτερική κωδικοποίηση ταυτίζεται με την εξωτερική και το σήμα `quad1` μετονομάζεται σε `block_4`. Τα περιεχόμενα των LUT αναγράφονται στον πίνακα 4.18.

Στη συνέχεια παράγονται τα `k-block` που αντιστοιχούν στον κάθε τύπο διαμόρφωσης. Τα 4 και 2 MSB των λέξεων `block_4` και `block_16` αντίστοιχα, γεμίζουν με μηδενικά bit πλεονασμού, ώστε να σχηματίσουν μία λέξη των 6 bit. Η λέξη `block_64` μένει ως έχει. Ένας MUX (MUX_1) λίγο πριν την έξοδο επιλέγει το έγκυρο `k-block` σύμφωνα με τον τρέχων τύπο διαμόρφωσης.

Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
0	0	0	0	16	6	32	48	48	54
1	1	1	1	17	7	33	49	49	55
2	4	2	8	18	14	34	56	50	62
3	5	3	9	19	15	35	57	51	63
4	3	4	3	20	5	36	51	52	53
5	2	5	2	21	4	37	50	53	52
6	7	6	11	22	13	38	59	54	61
7	6	7	10	23	12	39	58	55	60
8	12	8	24	24	30	40	40	56	46
9	13	9	25	25	31	41	41	57	47
10	8	10	16	26	22	42	32	58	38
11	9	11	17	27	23	43	33	59	39
12	15	12	27	28	29	44	43	60	45
13	14	13	26	29	28	45	42	61	44
14	11	14	19	30	21	46	35	62	37
15	10	15	18	31	20	47	34	63	36

(α) LUT_RC16

(β) LUT_RC64

Πίνακας 4.18: Περιεχόμενα των LUT_RC16 και LUT_RC64

4.11 Assembler [ASM]

4.11.1 Η Top-Level Όψη

Στο σχήμα 4.31 παρουσιάζεται το top-level διάγραμμα του Assembler. Ο Assembler, κατά μια έννοια, αποτελεί το κατοπτρικό module του Disassembler. Σκοπός του είναι η ένωση των k-blocks σε ομάδες των 8 bit προκειμένου να σχηματιστούν τα byte εξόδου του συστήματος. Η δουλειά που έχει να επιτελέσει είναι σαφώς πιο "εύκολη" και συνεκδοχικά, η πολυπλοκότητά του είναι σαφώς λιγότερη από αυτή του Disassembler. Ο κύριος λόγος είναι ότι, λόγω της θέσης του στην αλυσίδα του δέκτη, δεν υπάρχουν αυστηροί χρονικοί περιορισμοί για τη λειτουργία του. Παρ' όλα αυτά δεν είναι ένα τετριμμένο module και η υλοποίησή του -όπως θα φανεί παρακάτω- κρύβει τις δικές της δυσκολίες.

Τώρα, ο ωφέλιμος ρυθμός εισόδου (*input payload rate*) των δεδομένων είναι μεταβλητός και ανά περίπτωση ίσος με $[2, 4, 6] \frac{\text{bit}}{\text{datacycle}}$, ενώ ο ρυθμός εξόδου είναι $8 \frac{\text{bit}}{\text{datacycle}}$. Για την κατανομή των ωφέλιμων και των πλεονάζοντων (αδιάφορων) bit ισχύουν τα του πίνακα 4.1.

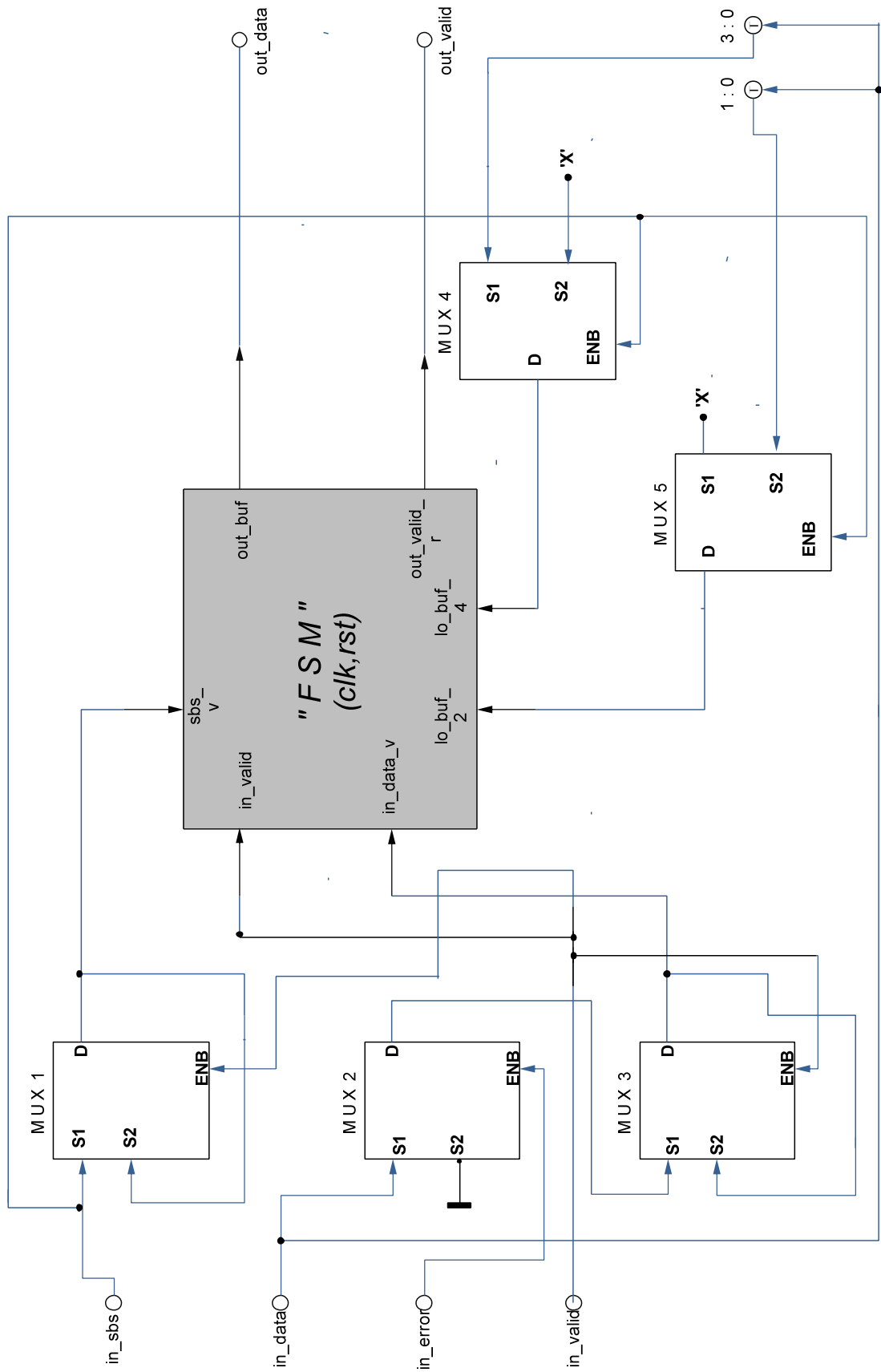
Πέρα από τα δεδομένα καθεαυτά (k-blocks), προκειμένου να συντελεστεί η διαδικασία της συνένωσης, το module χρειάζεται και την (κωδικοποιημένη) πληροφορία του τύπου διαμόρφωσης του τρέχοντος k-block την οποία λαμβάνει μέσω της θύρας εισόδου *in_sbs*.

4.11.2 Περιγραφή Κυκλωματικής Λειτουργίας

MUX-1

Σκοπός του MUX-1 είναι η διασφάλιση της εγκυρότητας του σήματος *sbs* (*in_sbs*) με σήμα ελέγχου το *in_valid*. Στην περίπτωση *in_valid* = 0 διατηρείται η προηγούμενη έγκυρη τιμή του *sbs*, οπότε μπορεί να ειπωθεί ότι ο πολυπλέκτης αυτός αποτελεί μια ιδιότυπη μορφή μανδαλωτή.

```
1 with in_valid select
2   sbs_v <= sbs when '1',
```



Σχήμα 4.31: ASM Top-Level διάγραμμα

```
3 sbs_v when others;
```

MUX-2

Ο πολυπλέκτης αυτός χειρίζεται την τιμή του σήματος *in_data_a* με σήμα ελέγχου το *in_error*. Η ύπαρξή του οφείλεται στην ανάγκη αποκλεισμού των (πιθανών) λανθασμένων εισόδων δεδομένων που έρχονται από το προηγούμενο module στην αλυσίδα (Symbol Demapper). Αυτά τα λάθη προκύπτουν από λανθασμένες αναγνωρίσεις συμβόλων στο στάδιο της αποδιαμόρφωσης κατά QAM. Σκοπός, λοιπόν, του πολυπλέκτη αυτού είναι ο αποκλεισμός αυτών των k-blocks (μέσω του μηδενισμού της τιμής του σήματος *in_data_a*) έτσι ώστε η διαδικασία της αποσφαλμάτωσης, κατά τον διασταυρούμενο έλεγχο του συστήματος με το μοντέλο MATLAB, να είναι αποδοτικότερη. Από καθαρά λειτουργική άποψη η παρουσία του δεν είναι απαραίτητη.

```
1 with in_error select
2   in_data_a <= (others=>'0') when '1',
3   in_data when others ;
```

MUX-3

Η λογική του πολυπλέκτη αυτού είναι παρόμοιας φύσης με αυτή του πολυπλέκτη MUX-1. Αποτελεί και αυτός έναν "μανδαλωτή εγκυρότητας" αυτή τη φορά για το σήμα *in_data_a* (*in_data*), με σήμα ελέγχου, φυσικά, το *in_valid*.

```
1 with in_valid select
2   in_data_v <= in_data_a when '1',
3   in_data_v when others;
```

Η διεργασία FSM - Μία άτυπη "μηχανή καταστάσεων"

Η διεργασία (process) FSM είναι αυτή που αναλαμβάνει να φέρει εις πέρας τον πυρήνα της λειτουργίας του Assembler. Ο πίνακας 4.19 παρουσιάζει συνοπτικά τις άτυπες "θύρες" εισόδου και εξόδου της διεργασίας αυτής.

Τώρα, προκειμένου να καταστεί πιο κατανοητή η λειτουργία της διεργασίας, αυτή θα αναλυθεί με όρους μιας άτυπης "μηχανής καταστάσεων". Στο Σχήμα 4.32 παρουσιάζεται η "μηχανή καταστάσεων" που κωδικοποιεί τον πυρήνα της λειτουργίας του Assembler.

Όπως φαίνεται στο διάγραμμα 4.32, υπάρχουν πέντε δυνατές "καταστάσεις" στις οποίες μπορεί να βρεθεί το κύκλωμα :

$$\{idle, less - than - eight, eight, ten, twelve\}$$

Το όνομα της κάθε κατάστασης (πλην της *idle*) αντικατοπτρίζει την τρέχουσα τιμή της μεταβλητής (variable) της διεργασίας, *sum*. Η τιμή της *sum*, τώρα, προκύπτει σε κάθε data cycle (*in_valid* = 1) από την ακόλουθη έκφραση :

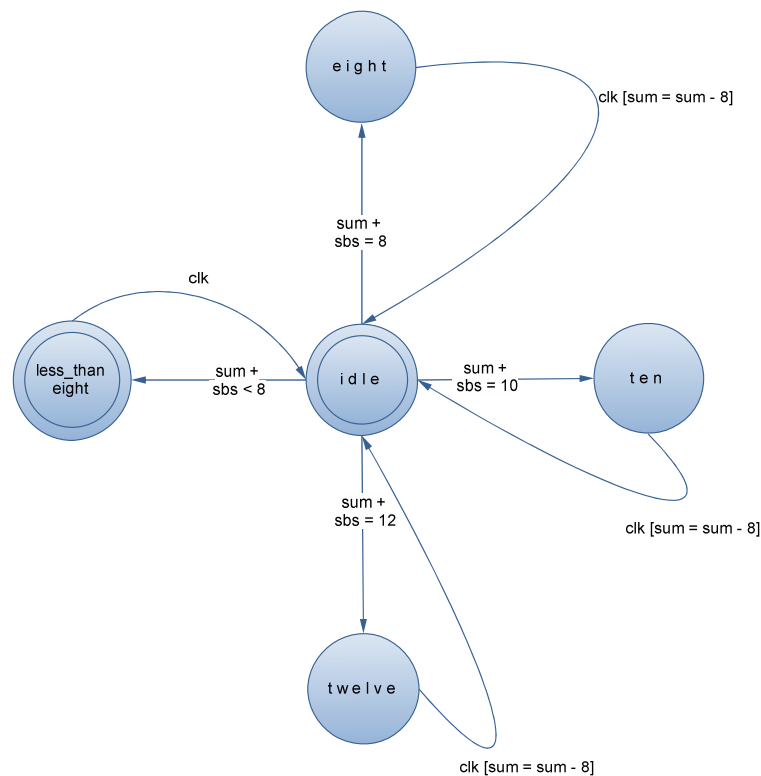
$$sum = sum + sbs_v$$

επομένως, η *sum* κρατάει το άθροισμα των διαδοχικών τιμών του *sbs_v* (*sbs*).³¹

³¹Υπενθυμίζεται, εδώ, ότι η τιμή του *sbs* είναι ανάλογη του τύπου διαμόρφωσης των k-blocks. Συγκεκριμένα : QAM-4 → 010₂ → 2 | QAM-16 → 100₂ → 4 | QAM-64 → 110₂ → 6. Φυσικά, η τιμή του *sbs* είναι ίση με την παράμετρο k των k-blocks, με άλλα λόγια, η δεκαδική τιμή του *sbs* αποτελεί το μήκος του υπό συνένωση συρμού bit (k-block).

Όνομα Θύρας	Τύπος	Περιγραφή
<i>sbs_v</i>	Είσοδος	Κωδικοποιημένη μορφή του τύπου διαμόρφωσης κατά QAM. Αναφέρεται στα τρέχοντα δεδομένα.
<i>in_valid</i>	Είσοδος	Υλοποιεί την εξυπηρέτηση του μηχανισμού frontpressure που δέχεται το module αυτό, από το upstream.
<i>in_data_v</i>	Είσοδος	Τα k-blocks των δεδομένων εισόδου.
<i>lo_buf_2</i>	Είσοδος	Ενταμιευτής περισσευόμενων bit (leftover-bits buffer), μεγέθους 2 bit.
<i>lo_buf_4</i>	Είσοδος	Ενταμιευτής περισσευόμενων bit (leftover-bits buffer), μεγέθους 4 bit.
<i>out_buf</i>	Εξοδος	Ενταμιευτής των byte εξόδου (output buffer).
<i>out_valid_r</i>	Εξοδος	Η registered εκδοχή του σήματος out_valid. Πιστοποιεί την εγκυρότητα των δεδομένων εξόδου και είναι ο φορέας του μηχανισμού frontpressure που ασκεί το module αυτό στα (ενδεχόμενα) module του downstream.

Πίνακας 4.19: "Θύρες" Ε/Ε της διεργασίας FSM



Σχήμα 4.32: Η άτυπη μηχανή καταστάσεων του Assembler

Το πέρασμα του κυκλώματος από τις καταστάσεις *ten* και *twelve* σηματοδοτείται με τη χρήση δύο σημαιών (flags), η τιμή των οποίων υπόκειται σε έλεγχο στην παρούσα κατάσταση προκειμένου να καθοριστει η συνέχεια. Συγκεκριμένα :

- Κατάσταση *ten*: • Σημαία : **ten**
- Κατάσταση *twelve*: • Σημαία : **twelve**

Εδώ, θα πρέπει να τονιστεί ότι απαραίτητη προϋπόθεση για τη μετάβαση από μια κατάσταση σε άλλη είναι να υπάρχουν έγκυρα δεδομένα εισόδου (*in_valid* = 1), δηλαδή, μεταβάσεις καταστάσεων προκύπτουν

μόνο σε data cycles και εφόσον $sum \geq 8$. Η επιλογή της τιμής 8 (eight) ως "κεντρικής" δεν είναι τυχαία, καθώς 8 bit σχηματίζουν ένα byte που είναι η δομή "συσκευασίας" των δεδομένων εξόδου.

Τώρα, ο καθαυτός μηχανισμός της μετάβασης από τις άλλες καταστάσεις (*eight*, *ten*, *twelve*) στην *less-than-eight*, υλοποιείται μέσω της έκφρασης

$$sum = sum - 8$$

όπου το 8 αντιπροσωπεύει τα οκτώ bit των διαφόρων k-blocks τα οποία έχουν ήδη ενωθεί σε byte και έχουν κατευθυνθεί στην έξοδο *out_data* του module.

Η κατάσταση *idle*

Σε αυτή την κατάσταση λαμβάνεται η απόφαση για την επόμενη κατάσταση που θα βρεθεί το κύκλωμα, με βάση την αποτίμηση της έκφρασης

$$sum := sum + sbs$$

Είναι μια virtual "αρχική κατάσταση", με την έννοια ότι το κύκλωμα δε στέκεται σε αυτή, αλλά δρομολογεί την επόμενη κατάσταση βάσει της τιμής του *sum* και μάλιστα, η μετάβαση γίνεται στον ίδιο κύκλο ρολογιού. Όλοι οι συνδυασμοί των (επιτρεπτών) μεταβάσεων από κατάσταση σε κατάσταση την περιλαμβάνουν ως ενδιάμεσο στάδιο.³²

Τώρα, πριν την αναλυτική περιγραφή των υπολοίπων καταστάσεων που μπορεί να βρεθεί το κύκλωμα, θα παρουσιαστούν με μια κάποια λεπτομέρεια δύο βοηθητικές δομές αποθήκευσης bit : Οι ενταμιευτές περισσευόμενων bit (left-over-bits buffers), *lo_buf_2* και *lo_buf_4*.

Οι "ενταμιευτές" *lo_buf_2* και *lo_buf_4*

Με εξαίρεση την περίπτωση όπου το τρέχον *sbs* είναι ίσο με 2, όπου η μόνη πιθανή μετάβαση είναι από την κατάσταση *less-than-eight* στην *eight* και επομένως -όπως θα φανεί παρακάτω- δεν υπάρχει κίνδυνος απώλειας δεδομένων, στις άλλες δύο περιπτώσεις ($sbs = 4$ και $sbs = 6$) ο κίνδυνος αυτός υπάρχει, στην περίπτωση όπου με την άθροιση και του τρέχοντος *sbs* η τιμή του *sum* προκύψει μεγαλύτερη από 8.

Πράγματι, έστω η περίπτωση όπου $sum = 4$ και (το τρέχον) $sbs = 6$. Εδώ τα συνολικά bit των k-blocks είναι 10 και αφαιρώντας τα 8 bit -που θα σχηματίσουν το byte εξόδου- περισσεύουν 2 bit τα οποία κινδυνεύουν να χαθούν αν δεν προβλεφτεί μια κατάλληλη δομή αποθήκευσης. Αυτά είναι τα λεγόμενα *περισσευόμενα bit* (left-over bits) και η δομή αποθήκευσης είναι οι ενταμιευτές *lo_buf_2* και *lo_buf_4* μεγέθους 2 και 4 bit, αντίστοιχα.³³

Ουσιαστικά, δηλαδή όποτε απαιτείται, οι buffers κρατούν ένα "αντίγραφο ασφαλείας" των bit της εισόδου τα οποία κινδυνεύουν να μείνουν "εκτός byte" σε περίπτωση που ο συνδυασμός του *sbs* και της τρέχουσας τιμής του *sum* δεν προκύψει βολικός (≤ 8). Όταν χρειαστεί, τα περιεχόμενά τους θα συνδυαστούν κατάλληλα με τα τότε τρέχοντα δεδομένα και θα προκύψει το byte εξόδου, χωρίς απώλεια κανενός bit.

Οι buffers αυτοί γεμίζουν δεδομένα μέσω των πολυπλεκτών MUX-4 και MUX-5 με τον τρόπο που φαίνεται παρακάτω (η αναφορά στη θέση των bit -πρώτα/τελευταία- θα γίνεται σύμφωνα με τη λογική *big endian*, δηλαδή το MSB θεωρείται το πρώτο bit) :

MUX-4

```
1 with in_sbs select
2   lo_buf_4 <= in_data_a(3 downto 0) when "110", --6
```

³²Στη συνέχεια του κειμένου δε θα αναφέρεται το πέρασμα του κυκλώματος από αυτή την κατάσταση, π.χ. η έκφραση "... γίνεται μετάβαση από την κατάσταση X στην κατάσταση Y μέσω της *idle* ($X \rightarrow idle \rightarrow Y$) ..." θα αναφέρεται απλώς ως "...γίνεται μετάβαση από την κατάσταση X στην Y ($X \rightarrow Y$)..."

³³Στο υπόλοιπο της περιγραφής αυτού του module, οι buffer αυτοί θα αναφέρονται και ως *lob-buffers*.

```
3 (others => 'X') when others; --2 OR 4
```

MUX-5

```
1 with in_sbs select
2   lo_buf_2 <= (others => 'X') when "010", --2
3   in_data_a(1 downto 0) when others; --4 OR 6
```

Αμφότεροι οι πολυπλέκτες έχουν ως σήμα-επιλογή το *sbs* και, ανάλογα με την τιμή του, συμβαίνουν τα εξής:

- *SBS* = 2
 - Και οι δύο buffer φορτώνουν με αδιάφορα bit
- *SBS* = 4
 - Ο *lo_buf_2* φορτώνει με τα 2 τελευταία bit των δεδομένων εισόδου
 - Ο *lo_buf_4* φορτώνει με αδιάφορα bit
- *SBS* = 6
 - Ο *lo_buf_2* φορτώνει με τα 2 τελευταία bit των δεδομένων εισόδου
 - Ο *lo_buf_4* φορτώνει με τα 4 τελευταία bit των δεδομένων εισόδου

Ο κύριος λόγος ύπαρξης των πολυπλεκτών (και όχι απλών συρμάτων), είναι η διευκόλυνση της διαδικασίας της αποσφαλμάτωσης (debugging).

Ο επικουρικός ενταμιευτής εξόδου *out_buf_next*

Στην πραγματικότητα, οι ενταμιευτές *lo_buf_2* και *lo_buf_4* είναι εικονικοί. Τα ομώνυμα σήματα του κυκλώματος είναι απλά σύρματα, που οδηγούν τα περισσευόμενα bit στον *out_buf_next*. Ο τελευταίος, αποτελεί τη κυκλωματική υλοποίηση της έννοιας των *lo_buf_2* και *lo_buf_4* και σκοπός του είναι να βοηθήσει στο σχηματισμό του μελλοντικού byte, έχοντας ως περιεχόμενο (μετά από κατάλληλη απόδοση) αυτό ενός εκ των lob-buffers. Το υπόλοιπο byte εξόδου θα γεμίσει με τα (τότε) τρέχοντα δεδομένα εισόδου (τόσα bit όσα χρειάζονται για να συμπληρωθεί).

Η κατάσταση *less-than-eight*

Τώρα, η κατάσταση *less-than-eight* αποτελεί, κατά μια έννοια, την πραγματική "αρχική κατάσταση" του κυκλώματος. Το κύκλωμα παραμένει σε αυτήν την κατάσταση για όσα data cycles το sum είναι μικρότερο από οκτώ ($sum < 8$). Το μεγαλύτερο διάστημα παραμονής σε αυτή την κατάσταση είναι 3 data cycles (τρία συνεχόμενα $sbs = 2$). Σε αυτή την κατάσταση δεν υπάρχουν έγκυρα δεδομένα εξόδου ($out_valid_r = 0$) καθώς τα υπάρχοντα k-blocks δεν είναι αρκετά για να σχηματιστεί ένα byte εξόδου, ενώ τα k-blocks εναποτίθενται, διαδοχικά, στον buffer εξόδου *out_buf*, ο οποίος φυσικά δεν καθίσταται ποτέ πλήρης για όσο διάστημα το κύκλωμα παραμένει σε αυτή την κατάσταση.

Τα παραπάνω συμβαίνουν στην αρχική κατάσταση της λειτουργίας του συστήματος του δέκτη και κάθε φορά που γίνεται επιστροφή του κυκλώματος σε αυτή την κατάσταση από την κατάσταση *eight*.³⁴

Η περίπτωση *twelve = 1*

Η προηγούμενη κατάσταση του κυκλώματος ήταν η *twelve*. Μπορεί να συμβεί μόνο αν $sbs = 2$, αφού

³⁴Γεγονός το οποίο είναι προφανές, αφού $8 - 8 = 0$

$12 - 8 + 2 = 6 < 8$. Τα 6 πρώτα bit του byte εξόδου προέρχονται από : τα 4 πρώτα bit του *out_buf_next* και τα 2 τελευταία (ωφέλιμα) bit των δεδομένων εισόδου (*in_data_v*).

Η τιμή του *sum* παραμένει η ίδια, το σήμα *out_valid_r* παραμένει ίσο με 0 και πέφτει η σημαία *twelve*. Επόμενες καταστάσεις μπορεί να είναι όλες (πλην της *less-than-eight*).

Η περίπτωση *ten = 1*

Η προηγούμενη κατάσταση του κυκλώματος ήταν η *ten*. Μπορεί να συμβεί αν *sbs = 4* ή *sbs = 6*. Σε περίπτωση που *sbs = 2* (*sbs = 4*), τα 4(6) πρώτα bit του byte εξόδου προέρχονται από : τα 2(2) πρώτα bit του *out_buf_next* και τα 2(4) τελευταία (ωφέλιμα) bit των δεδομένων εισόδου (*in_data_v*).

Η τιμή του *sum* παραμένει η ίδια, το σήμα *out_valid_r* παραμένει ίσο με 0 και πέφτει η σημαία *ten*. Όταν *sbs = 2*, επόμενες καταστάσεις μπορεί να είναι οι *less-than-eight*, *eight* και *ten*. Όταν *sbs = 4*, επόμενες καταστάσεις μπορεί να είναι οι *eight*, *ten* και *twelve*.

Η κατάσταση *eight*

Η κανονική περίπτωση

Σε αυτή την κατάσταση περιέρχεται το κύκλωμα όταν έχει παρουσιαστεί μια "βολική" ακολουθία *sbs*, τέτοια ώστε το άθροισμά της να είναι ίσο με οκτώ. Παραδείγματα αποτελούν οι ακολουθίες (2,2,2,2)-(2,4,2)-(4,4)-(6,2) κλπ. Η προηγούμενη κατάσταση ήταν η *less-than-eight*. Σε αυτή την περίπτωση το byte σχηματίζεται "αυτόματα" από τα υπάρχοντα *k*-blocks και προωθείται στην έξοδο του κυκλώματος.

Η τιμή του *sum* γίνεται ίση με μηδέν και το σήμα *out_valid_r* τίθεται ίσο με 1. Επόμενη κατάσταση είναι η *less-than-eight*.

Η περίπτωση *twelve = 1*

Η προηγούμενη κατάσταση του κυκλώματος ήταν η *twelve*. Μπορεί να συμβεί μόνο αν *sbs = 4*, αφού $12 - 8 + 4 = 8$. Το byte εξόδου σχηματίζεται από : τα 4 πρώτα bit του *out_buf_next* και τα 4 τελευταία (ωφέλιμα) bit των δεδομένων εισόδου (*in_data_v*).

Η τιμή του *sum* μηδενίζεται, το σήμα *out_valid_r* παραμένει ίσο με 1 και πέφτει η σημαία *twelve*. Επόμενη κατάσταση είναι η *less-than-eight*.

Η περίπτωση *ten = 1*

Η προηγούμενη κατάσταση του κυκλώματος ήταν η *ten*. Μπορεί να συμβεί μόνο αν *sbs = 6*, αφού $10 - 8 + 6 = 8$. Το byte εξόδου σχηματίζεται από : τα 2 πρώτα bit του *out_buf_next* και τα 6 ωφέλιμα bit των δεδομένων εισόδου (*in_data_v*).

Η τιμή του *sum* μηδενίζεται, το σήμα *out_valid_r* παραμένει ίσο με 1 και πέφτει η σημαία *ten*. Επόμενη κατάσταση είναι η *less-than-eight*.

Η κατάσταση *ten*

Προκύπτει όταν *sum = 10*. Ανάλογα με το ποιά ήταν η προηγούμενη κατάσταση προκύπτουν δύο περιπτώσεις.

Η περίπτωση *twelve = 1*

Η προηγούμενη κατάσταση του κυκλώματος ήταν η *twelve*. Μπορεί να συμβεί μόνο αν *sbs = 6*, αφού $12 - 8 + 6 = 10$. Το byte εξόδου σχηματίζεται από : τα 4 πρώτα bit του *out_buf_next* και τα 4 πρώτα bit των δεδομένων εισόδου (*in_data_v*). Τα υπόλοιπα 2 bit της εισόδου εκχωρούνται -μέσω του *lo_buf_2*- στον *out_buf_next*, για μελλοντική χρήση.

Η νέα τιμή του *sum* είναι 2 ($10 - 8 = 2$), το σήμα *out_valid_r* παραμένει ίσο με 1 και σηκώνεται η σημαία *ten* ενώ πέφτει η σημαία *twelve*. Επόμενη κατάσταση είναι η *less-than-eight*.

Η κανονική περίπτωση

Η προηγούμενη κατάσταση του κυκλώματος ήταν η *less-than-eight*. Μπορεί να συμβεί αν $sbs = 4$ ή $sbs = 6$. Σε περίπτωση που $sbs = 4$, τα 2 τελευταία bit του byte εξόδου προέρχονται από τα 2 μεσαία (πρώτα ωφέλιμα) bit των δεδομένων εισόδου (*in_data_v*). Σε περίπτωση που $sbs = 6$, τα 4 τελευταία bit του byte εξόδου προέρχονται από τα 4 πρώτα bit των δεδομένων εισόδου (*in_data_v*). Και στις δύο περιπτώσεις τα υπόλοιπα 2 ωφέλιμα bit των δεδομένων εισόδου εκχωρούνται -μέσω του *lo_buf_2*- στον *out_buf_next*, για μελλοντική χρήση.

Η νέα τιμή του *sum* είναι 2 ($10 - 8 = 2$), το σήμα *out_valid_r* τίθεται ίσο με 1. Επόμενη κατάσταση είναι η *less-than-eight*.

Η κατάσταση *twelve*

Το κύκλωμα μεταβαίνει σε αυτή την κατάσταση μόνο όταν ισχύουν ταυτόχρονα οι συνθήκες $sum = 6$ και $sbs = 6$. Σε αυτή την περίπτωση στα 6 υπάρχοντα bit του buffer εξόδου *out_buf* προστίθενται τα 2 πρώτα bit των δεδομένων εισόδου (*in_data_v*) και τα υπόλοιπα 4 εκχωρούνται -μέσω του *lo_buf_4*- στον *out_buf_next*, για μελλοντική χρήση.

Η νέα τιμή του *sum* είναι 4 ($12 - 8 = 4$), το σήμα *out_valid_r* τίθεται ίσο με 1 και σηκώνεται η σημαία *twelve*. Επόμενη κατάσταση είναι η *less-than-eight*.

Κεφάλαιο 5

Αποτίμηση Συστήματος

Μέχρι τώρα έχει παρουσιαστεί με λεπτομέρεια η αρχιτεκτονική και η εσωτερική οργάνωση του συστήματος. Σε αυτό το κεφάλαιο, θα γίνει η αποτίμησή του από τηλεπικοινωνιακής και λειτουργικής άποψης. Αφενός, εξηγείται ο τρόπος με τον οποίο διενεργήθη ο έλεγχος ορθότητας λειτουργίας του πομποδέκτη, χωρίς την παρουσία καναλιού (wire channel) και, αφετέρου, θα παρουσιαστούν τα μεγέθη που ποσοτικοποιούν την απόδοση του συστήματος και προκύπτουν από την προσομοίωση της διαδικασίας λήψης μέσα από διάφορα τηλεπικοινωνιακά κανάλια. Αυτή είναι η φάση της αποτίμησης και αφορά κυρίως στο δέκτη, καθώς εκεί εισάγονται τα περισσότερα σφάλματα λόγω της φύσης της αποδιαμόρφωσης.

5.1 Λειτουργικός Έλεγχος Συστήματος

5.1.1 Το Testbench

Για τον έλεγχο της ορθής λειτουργίας ενός κυκλώματος γίνεται λειτουργική προσομοίωση αυτού, με χρήση ειδικού λογισμικού προσομοίωσης. Το τελευταίο, «εκτελεί» το κύκλωμα δεδομένης μιας διέγερσης εισόδου και υπολογίζει όλα τα ενδιάμεσα σήματα και τις εξόδους. Τα σήματα εισόδου/εξόδου, αλλά και οποιοδήποτε εσωτερικό σήμα δύναται να απεικονιστούν σε ένα διάγραμμα κυματομορφών.¹ Στην προκειμένη περίπτωση οι προσομοιώσεις πραγματοποιήθηκαν με το περιβάλλον εξομοίωσης *Modelsim 6.5f* της εταιρίας Modeltech.

Για την παραγωγή της διέγερσης του κυκλώματος γράφεται ένα βοηθητικό «κύκλωμα» που καλείται testbench. Το testbench, παρόλο που είναι γραμμένο σε VHDL, δεν περιγράφει κάποιο συνθέσιμο κύκλωμα, αλλά αποτελεί ένα πρόγραμμα διεπαφής του κυκλώματος με το λογισμικό προσομοίωσης. Οι δομές ελέγχου και οργάνωσης του κώδικα είναι οι κλασικές που απαντώνται σε ένα VHDL κύκλωμα. Το top level component που αποτελεί το κύκλωμα, χρησιμοποιείται για τη δημιουργία ενός στιγμιότυπου. Στη συνέχεια γράφεται (συντρέχων ή ακολουθιακός) κώδικας VHDL που διεγείρει τις εισόδους του κυκλώματος και ενδεχομένως καταγράφει σε κάποιο αρχείο την έξοδο για περαιτέρω ανάλυση. Επιπλέον, παρέχονται στον προγραμματιστή βοηθητικές συναρτήσεις για την εκτέλεση διαφόρων λειτουργιών όπως: ανάγνωση και εγγραφή δεδομένων σε αρχεία, εκτύπωση διαγνωστικών μηνυμάτων στην κονσόλα, παρατήρηση εσωτερικών σημάτων όλου του κυκλώματος, κ.α.

Η κύρια μέθοδος ελέγχου της λειτουργίας του κυκλώματος είναι η παρατήρηση του διαγράμματος κυματομορφών. Οι εισοδοί, οι έξοδοι και οποιοδήποτε εσωτερικό σήμα επιθυμεί κανείς, απεικονίζονται σε ένα γράφημα χρονικής εξέλιξής τους. Τα περισσότερα προβλήματα εντοπίζονται μέσα από προσεκτική παρατήρηση των τιμών των σημάτων, αλλά και των μεταξύ τους σχέσεων. Το σχήμα των παλμών και η

¹ Σε αυτό το στάδιο η προσομοίωση είναι σε επίπεδο καταχωρητών (*Register Transfer Level - RTL*), οι οποίοι, μάλιστα, θεωρούνται ιδανικοί. Η εξακρίβωση της ορθής κυκλωματικής λειτουργίας θα γίνει σε μεταγενέστερο στάδιο (*timing analysis*), όπου εκεί εισάγονται πιο ρεαλιστικά μοντέλα για τους καταχωρητές και τις λογικές πύλες.

διάρκειά τους είναι επίσης χαρακτηριστικά που βοηθούν στην αποσφαλμάτωση (debugging).

Η διέγερση των εισόδων του κυκλώματος, μπορεί να κωδικοποιηθεί στο testbench ή να διαβαστεί από αρχείο. Σχετικά απλά σήματα, όπως για παράδειγμα παλμοί, μπορούν να δημιουργηθούν σύμφωνα με ένα πρότυπο, ή απλά να κωδικοποιηθούν οι προκαθορισμένες τιμές τους.² Ωστόσο σε κυκλώματα που επιτελούνται σύνθετες μαθηματικές πράξεις ή υπάρχουν πολύπλοκες δομές ελέγχου, είναι δύσκολο να κωδικοποιηθεί η είσοδος με το "χέρι" και να επαληθευτεί η έξοδος με το "μάτι". Η συνήθης πρακτική για τον έλεγχο τέτοιου είδους κυκλωμάτων, είναι με τη χρήση διεργασιών (processes) που διαβάζουν την είσοδο από αρχείο και γράφουν την έξοδο σε άλλο αρχείο. Το αρχείο εισόδου μπορεί να γράφεται από κάποιο πρόγραμμα και το αρχείο εξόδου μπορεί να διαβάζεται από άλλο πρόγραμμα, το οποίο εκτελεί συστηματικά διάφορους ελέγχους ορθότητας.

5.1.2 Έλεγχος Μονάδας (Unit Testing)

Σε κάθε υποσύστημα, προτού αυτό εισαχθεί στην αλυσίδα του κυκλώματος, πραγματοποιείται διεξοδικός λειτουργικός έλεγχος, είτε με προσεκτική παρατήρηση των κυματομορφών, είτε με τη χρήση εξειδικευμένων προγραμμάτων MATLAB, όταν η πολυπλοκότητα είναι απαγορευτική για τον έλεγχο με το "μάτι". Για κάθε υποσύστημα, υπάρχει ξεχωριστό testbench. Η πρακτική αυτή ονομάζεται έλεγχος μονάδας (unit testing) και βοηθάει στην έγκαιρη αποσφαλμάτωση κατά τη διάρκεια της σχεδίασης. Ο σκοπός είναι να προβλεφθούν όσο το δυνατόν περισσότεροι συνδυασμοί μεταξύ των σημάτων εισόδου, για την απόκτηση μεγαλύτερης σιγουριάς όσον αφορά τη λειτουργική ορθότητα. Αν ένα πρόβλημα δεν αντιμετωπιστεί κατά το unit testing είναι πιο δύσκολο να ανακαλυφτεί στον έλεγχο συστήματος. Ωστόσο η διαδικασία δεν είναι απαραίτητα γραμμική. Αν εντοπιστεί κάποιο πρόβλημα στον έλεγχο συστήματος, ο σχεδιαστής μπορεί να επιστρέψει στο ύποπτο module και να διενεργήσει unit testing (με σκοπό να ανακαλυφθεί και εξαλειφθεί το πρόβλημα) απομονώνοντάς το από το υπόλοιπο σύστημα.

Βοηθητικά script ελέγχου (check scripts) στο MATLAB έχουν γραφτεί για τις εξής μονάδες (module):

- Time Synchronizer
- Frequency Compensator
- Channel Estimator
- Symbol Demapper

Για τα υπόλοιπα, απλούστερης φύσης, module η λειτουργική ορθότητα επαληθεύτηκε μέσω παρατήρησης των κυματομορφών τους.

Τα Αριθμητικά Μοντέλα

Για τον έλεγχο της ορθότητας των μαθηματικών πράξεων που επιτελούνται σε διάφορα (απαιτητικά από υπολογιστικής απόψεως) module, γράφτηκαν τα αντίστοιχα μοντέλα αριθμητικής στο MATLAB. Κάνοντας χρήση του *Fixed Point Toolbox* οι πράξεις γίνονται ακριβώς με τον ίδιο τρόπο όπως στο κύκλωμα VHDL, δηλαδή ακολουθώντας τα ίδια βήματα και χρησιμοποιώντας τις ίδιες ακρίβειες εργασίας (working precisions). Με αυτόν τον τρόπο, η ανάλυση σφάλματος και οι τροποποιήσεις γίνονται στο MATLAB, επιταχύνοντας κατά πολύ τη διαδικασία σχεδίασης. Επίσης, με τη βοήθεια των μοντέλων αυτών, γίνεται και η βελτιστοποίηση των ακριβειών εργασίας για την υλοποίηση διαφόρων συναρτήσεων.

Από τη στιγμή που θα επιτευχθεί η σωστή λειτουργία με την επιθυμητή ακρίβεια, η υλοποίηση μεταφέρεται σε VHDL και προσαρτάται στον πηγαίο κώδικα του module. Στη συνέχεια, με χρήση των αντιστοιχών

²Συνήθως με ακολουθιακό κώδικα, ως ένα process με μία εντολή case ή if.

check script, επιτελείται ο έλεγχος ορθής λειτουργίας του VHDL κυκλώματος, έχοντας σαν αναφορά το μοντέλο MATLAB.

Αριθμητικά μοντέλα MATLAB έχουν γραφτεί για τα εξής module:

- Frequency Compensator
- Channel Estimator
- Argument Computer

5.1.3 Έλεγχος Πλήρους Συστήματος (System Testing)

Απο τη στιγμή που ολοκληρώνεται η κατασκευή ενός module και περατώνεται ο λειτουργικός του έλεγχος, επόμενο στάδιο είναι η προσάρτησή του στη συστημική αλυσίδα, στην προβλεπόμενη για αυτό θέση. Αυτή η διαδικασία είναι πιο σύνθετη απ' ό,τι, αρχικά, μπορεί να φανταστεί κανείς καθώς, η συστημική συμπεριφορά ενός module όταν αυτό αποτελέσει έναν κρίκο της αλυσίδας του συστήματος, μπορεί να επηρεαστεί και να αποκλίνει της αναμενόμενης με τρόπους που είναι δύσκολο να προβλεφτούν από το σχεδιαστή κατά τα στάδια της αρχικής ανάπτυξής του. Μάλιστα, όσο πιο πολύπλοκη είναι η δομή και η λειτουργία που επιτελεί το module, τόσο αυξάνουν οι πιθανότητες να παρουσιαστεί κάποια απόκλιση κατά την τοποθέτησή του στη συστημική αλυσίδα.

Γι' αυτό το λόγο επιβάλλεται η διενέργεια αυστηρών λειτουργικών ελέγχων στην συστημική αλυσίδα, προκειμένου να διασφαλιστεί η ορθότητα και η τελεσφορία του συστήματος. Το σύνολο αυτών των ελέγχων ονομάζεται *Έλεγχος Συστήματος (System Testing)*. Μία βασική δομή που επιτρέπει την επιτυχημένη και αποδοτική διενέργειά του είναι ένα πλήρες μοντέλο του συστήματος (πομπός και δέκτης) κατασκευασμένο στο περιβάλλον *MATLAB* (Μοντέλο *MATLAB*).

Το Μοντέλο *MATLAB*

Για την απόκτηση ενός πλαισίου αναφοράς, έχει κατασκευαστεί μοντέλο του συστήματος στο *MATLAB*, τόσο για τον πομπό, όσο και για τον δέκτη. Το μοντέλο, από λειτουργική και αλγοριθμική άποψη, είναι όσο το δυνατόν πιο συνεπές σε σχέση με το κύκλωμα. Ωστόσο οι αριθμητικές πράξεις εκτελούνται με *double precision*, καθώς δεν ενδιαφέρει η πλήρης αντιστοίχιση των ακριβειών του μοντέλου και του κυκλώματος. Αυτή η εργασία έχει πραγματοποιηθεί στο στάδιο του unit testing. Δηλαδή το μοντέλο αποτελεί περισσότερο ένα *λειτουργικό πρότυπο (functional prototype)* για το κύκλωμα, παρά μία πιστή μεταφορά του στο περιβάλλον *MATLAB*.

Assets - Parameters

Όπως έχει αναφερθεί στο Κεφάλαιο 3, υπάρχουν κάποια στοιχεία που μπορούν να αλλάξουν σε διαφορετικά στιγμιότυπα του κυκλώματος, οι παράμετροι και τα asset. Ο μηχανισμός *asset-dispatching* είναι υπεύθυνος για το συγχρονισμό των τιμών των παραμέτρων και των asset που έχουν διασπαρεί σε διάφορα σημεία μεταξύ μοντέλου και κυκλώματος. Οι παράμετροι και τα asset, τροποποιούνται κεντρικά (σε ένα σημείο) πριν από την εκτέλεση των δοκιμών και διανέμονται στα κατάλληλα σημεία στο κύκλωμα και στο μοντέλο. Στη συνέχεια πραγματοποιούνται οι αντίστοιχες προσομοιώσεις.

5.1.4 Πομπός

Το πρώτο βήμα για το χτίσιμο του πλαισίου αναφοράς είναι η δημιουργία ενός μοντέλου για τον πομπό. Επιπλέον, το μοντέλο του πομπού χρησιμεύει για την παραγωγή των δεδομένων εισόδου που τροφοδο-

τούνται στο δέκτη. Οι λειτουργίες της διαμόρφωσης υλοποιούνται σε ξεχωριστά στάδια, έτσι ώστε καθένα να αντιστοιχεί σε ένα module του TX datapath. Το κάθε στάδιο είναι ένα ξεχωριστό υποπρόγραμμα MATLAB το οποίο μπορεί να εκτελείται αυτόνομα. Η έξοδος του γράφεται σε αρχείο κειμένου, μαζί με βοηθητικούς δείκτες που αριθμούν τα διάφορα samples. Με αυτόν τον τρόπο, το σύστημα γίνεται εσωτερικά παρατηρήσιμο και διευκολύνεται η διαδικασία της αποσφαλμάτωσης.

Τα στάδια διαμόρφωσης του μοντέλου είναι:

1. Διαμόρφωση κατά QAM • Subsymbol Mapping
2. Προσθήκη Training Symbols • Training symbol addition
3. Διαμόρφωση κατά OFDM • OFDM Modulation

Για την παραγωγή δεδομένων εισόδου υπάρχει ένα script MATLAB που δημιουργεί τυχαίες ακολουθίες byte. Ένα άλλο script αναλαμβάνει την διάθεση των δεδομένων εισόδου στο κύκλωμα και στο μοντέλο, εξασφαλίζοντας το συγχρονισμό μεταξύ τους.

Οι τηλεπικοινωνιακές παράμετροι του συστήματος αναγράφονται στους Πίνακες (2.2) και (2.3) του Κεφαλαίου 2 και συνοψίζονται εδώ.

Παράμετροι διαμόρφωσης

```

1 * OFDM transmitter parameters
2 Useful symbol size      : 128 samples
3 Cyclic prefix size     : 32 samples
4 Total symbol size      : 160 samples
5 Frame data part size   : 7 symbols
6 Frame size             : 10 symbols
7 Constel scaling int16 pcnt : 48 %
8 TX postFFT scaling     : 1/16

```

Στη συνέχεια παρουσιάζονται διάφορα χαρακτηριστικά μεγέθη του σήματος εξόδου του πομπού, όπως έχουν υπολογιστεί από το μοντέλο.

Στατιστικά Διαμόρφωσης

```

1 * QAM-4 statistics
2 -----
3 Symbol covarage       : 100.0 %
4 Symbol count          : 8064
5 Byte count            : 2016.0
6
7 * General statistics
8 -----
9 Total symbol count    : 8064
10 Total byte count     : 2016.0
11
12 * Frame statistics
13 -----
14 # written frames     : 9
15 # written ofdm symbols : 90
16 # written subsymbols : 11520

```

Ποιοτικά χαρακτηριστικά σήματος πομπού

```

1 * Scaled saturated statistics

```



```

2 -----
3 Peak (amplitude) : 29081+29960i
4 Average (amplitude) : 6376+6126i
5 RMS (amplitude) : 8004+7768i
6 PAPR (signal) : 9.53 (9.79 dB)
7
8 * Int 16 statistics
9 -----
10 No of truncated values(re+im) : 0 (0 %)
11 Dynamic Range Utilization : 24.1 % (avg for 2 DACs)
12 Saturation pwr loss : 0.00 %
13
14 * Quantization statistics
15 -----
16 Max noise floor (i16) : -98.09 dB
17 Avg noise floor (sig) : -88.73 dB (avg for 2 DACs)

```

Παρατηρήσεις

Εδώ, είναι αξιοσημείωτο το σχετικά χαμηλό rms πλάτος του σήματος: 7886 (8004 για την I και 7768 για την Q συνιστώσα). Αυτό αντιστοιχεί κατά μέσο όρο στο 24.1 % του διαθέσιμου δυναμικού εύρους σε DAC με εύρος 16 bit. Η rms τιμή του πλάτους καθορίζεται από τους συντελεστές προ και μετά FFT κλιμάκωσης και η επιλογή τους έγινε έτσι ώστε, να αποφευχθεί πλήρως το clipping λόγω του υψηλού PAPR. Το κατώφλι ψηφιακού θορύβου που εισάγεται λόγω κβαντισμού, για τη συγκεκριμένη rms τιμή, είναι στα -88.7 dB.³ Αυτή η τιμή είναι περίπου 10 dB υψηλότερη από το ελάχιστο κατώφλι ψηφιακού θορύβου για 16 bit integer αριθμητική που είναι στα -98 dB, ωστόσο παραμένει εξαιρετικά χαμηλή σε σχέση με άλλες πηγές θορύβου που εισάγονται από ένα τυπικό κανάλι.

Script Ελέγχου

Ο έλεγχος της λειτουργικής ορθότητας πραγματοποιείται από ένα script ελέγχου (check script) στο MATLAB. Η έξοδος του μοντέλου συγκρίνεται με αυτή του κυκλώματος, όπως αυτή έχει παραχθεί μέσω λειτουργικής προσομοίωσης. Γράφεται ένα αρχείο διασταυρούμενου ελέγχου (cross-checking data-file), όπου μεταξύ άλλων σημειώνονται τα σφάλματα ανά sample. Εν τέλει υπολογίζεται το μέσο απόλυτο σφάλμα, ως μέτρο εγκυρότητας και επίδοσης του πομπού.

Η διαδικασία για τον έλεγχο του πομπού συνίσταται στα εξής:

1. Τροποποίηση των παραμέτρων ή/και των asset.
2. Παραγωγή δοκιμαστικών δεδομένων
3. Διάθεση των asset και των παραμέτρων (Asset Dispatching).
4. Διαμόρφωση δεδομένων στο μοντέλο.
5. Διαμόρφωση δεδομένων στο κύκλωμα (μέσω λειτουργικής προσομοίωσης).
6. Εκτέλεση του check script και αποτίμηση αποτελεσμάτων.

Αποτίμηση

Διαγνωστικά μυνήματα ελέγχου πομπού

³ Περισσότερα για τον ψηφιακό θόρυβο στην § 2.1.6.

```

1 * CHECK TX script started.
2 Mean |error| : 1.0458
3 Var error : 1.5482
4 ** Data mismatch between VHDL and MATLAB TX
5 * # error samples : 88 / 14240
6 * VDHL TX is 0.618% erroneous.
7 * And VHDL samples are delayed 0 samples referring to MATLAB.
8 Cross annotated data written
9 * CHECK TX script ended.

```

Διαμορφώνοντας 9 frames (90 OFDM symbols) το μέσο απόλυτο σφάλμα είναι 1.046, πρακτικά αμελητέο δεδομένου ότι η rms τιμή του σήματος εξόδου είναι 7886. Επιπλέον, μόλις το 0.62% των διαμορφωμένων samples διαφέρει από αυτά του μοντέλου. Η προέλευση του σφάλματος είναι ο αλγόριθμος FFT, καθώς στο κύκλωμα επιτελείται με *fixed point* αριθμητική, ενώ στο μοντέλο με *double precision*.

5.1.5 Δέκτης

Για τον έλεγχο του δέκτη δημιουργήθηκε το αντίστοιχο μοντέλο στο MATLAB. Ο κύριος σκοπός του είναι η δημιουργία σημάτων αναφοράς για τα διάφορα στάδια του δέκτη, ώστε να επαληθευτεί η λειτουργική του ορθότητα. Οι λειτουργίες που επιτελούνται από τα διακριτά module στο datapath του δέκτη, έχουν υλοποιηθεί ως ξεχωριστά υποπρογράμματα. Η έξοδος κάθε σταδίου γράφεται σε αρχείο κειμένου, μαζί με βοηθητικούς δείκτες αρίθμησης (ανά frame, symbol, subsymbol, κλπ), ώστε να διευκολύνεται η αποσφαλμάτωση και η αξιολόγηση του συστήματος.

Τα στάδια αποδιαμόρφωσης του μοντέλου είναι:

1. Συγχρονισμός στο χρόνο • Time Synchronization
2. Αντιστάθμιση Συχνότητας • Frequency Compensation
3. Αποδιαμόρφωση OFDM • OFDM Demodulation
4. Εκτίμηση και Ισοστάθμιση Καναλιού • Channel Estimation, Equalization
5. Αποδιαμόρφωση QAM • Symbol Demapping
6. Συναρμολόγηση Byte Εξόδου • Byte Assembly

Οι τηλεπικοινωνιακές παράμετροι του συστήματος αναγράφονται στους Πίνακες (2.2) και (2.3) του Κεφαλαίου 2 και συνοψίζονται εδώ.

Παράμετροι αποδιαμόρφωσης

```

1 * OFDM receiver parameters
2 Useful symbol size : 128 samples
3 Cyclic prefix size : 32 samples
4 Total symbol size : 160 samples
5 Frame data part size : 7 symbols
6 Frame data part size : 1120 samples
7 Frame size : 10 symbols
8 Frame size : 1600 samples
9 Metric search span : 60 samples
10 Metric threshold pcnt : 61 %
11 Constel scaling int16 pcnt : 48 %
12 RX postFFT scaling : 1/8

```

Script Ελέγχου

Υπεύθυνο για το λειτουργικό έλεγχο του δέκτη είναι ένα σύνολο από script ελέγχου (check scripts). Η βασική του λειτουργία είναι η σύγκριση της εξόδου του κυκλώματος με αυτή του μοντέλου, ενώ ταυτόχρονα γίνεται σύγκριση της εξόδου του *Symbol Demapper* και του *Time Synchronizer*.

Καταληκτικό στάδιο είναι τα αντίστοιχα αρχεία διασταυρούμενου ελέγχου (cross check) και η εκτύπωση διαγνωστικών μηνυμάτων. Σε αυτό το στάδιο δε μεσολαβεί πραγματικό κανάλι, αλλά η είσοδος του δέκτη συνδέεται απευθείας στην έξοδο του πομπού (wire channel). Ως είσοδος χρησιμοποιείται το OFDM σήμα που έχει διαμορφωθεί από το μοντέλο του πομπού, έτσι ώστε να αποτραπεί η παρουσία σφαλμάτων που προέρχονται από την πεπερασμένη ακρίβεια του κυκλώματος.

Αποτίμηση

Διαγνωστικά μηνύματα ελέγχου δέκτη

```

1 * CHECK RX script started.
2 * Checking time sync output
3   Mean |error|: 0
4 * Check VHDL sof points.
5   * SYNC SOF points are correct!
6 * Check VHDL args.
7   Delta f      : 0.00781250
8   Mean |error| : 0.00000243
9   Mean error   : -0.00000243
10  Var error    : 0.00000000
11 ** ARG mismatch between VHDL and MATLAB RX
12   # error argz : 9 / 9
13   VDHL RX argz are 100% erroneous.
14   Cross annotated data written
15 * Checking time sync output ended
16 * Checking assembled output
17   Mean |error|: 0
18   * VHDL assembler output is correct!
19 * Demodulation stats:
20 Err bits: 0 / 16128
21 BER: 0
22 Mean BIT errors: 0
23 * VHDL RX output is correct!
24 Cross annotated data written
25 * CHECK RX script ended.
```

Αποδιαμορφώθηκαν τα 9 OFDM frames που μεταδόθηκαν, φέροντας 2016 bytes ή 16128 bits ωφέλιμης πληροφορίας. Τα αποτελέσματα των παραπάνω ελέγχων έδειξαν ότι το κύκλωμα του δέκτη λειτουργεί σύμφωνα με το πρότυπο (μοντέλο) όσων αφορά την τελική έξοδο, έχοντας BER ίσο με μηδέν.

Ωστόσο παρατηρείται μία απόκλιση στην εκτίμηση ολίσθησης συχνότητας, όπου παρουσιάζεται μέσο σφάλμα -0.00243×10^{-3} . Η διακύμανση είναι 0, πράγμα που σημαίνει πως όλα τα frame έχουν το ίδιο σφάλμα όπως ήταν αναμενόμενο, καθώς δεν μετέχει καμία στοχαστική διαδικασία παραμόρφωσης του σήματος εισόδου. Η προέλευσή του σφάλματος είναι η ακρίβεια των υπολογισμών στο module ARG που βρίσκεται μέσα στον *Time synchronizer*. Όπως έχει αναφερθεί στην § 4.6.6 υπάρχει ένα εγγενές σφάλμα υπολογισμού ($\hat{\phi}_{err}$) της εκτίμησης της διαφοράς φάσης και ισχύει $|\hat{\phi}_{err}| < 0.00171$. Για το αντίστοιχο σφάλμα της εκτίμησης συχνότητας ισχύει $|\hat{\Psi}_{err}| < 0.00425 \times 10^{-3}$ σε μονάδες κανονικοποιημένων hertz.

Ισοδύναμα το σφάλμα εκτίμησης συχνότητας είναι 0.054% της απόστασης μεταξύ των subcarrier. Μετά από ανάλυση προέκυψε πως η συνεισφορά του σφάλματος αυτού στον εσωτερικό θόρυβο (*INNOISE*) είναι αμελητέα.

5.2 Τηλεπικοινωνιακή Αποτίμηση Δέκτη

Από το σημείο αυτό και μετά, προκειμένου να διερευνηθεί η τηλεπικοινωνιακή επίδοση του δέκτη, το σήμα εισόδου θα περάσει μέσα από κανάλια διαφόρων χαρακτηριστικών, διαδικασία η οποία αποτελεί το θέμα του υπόλοιπου κεφαλαίου.

5.2.1 Κανάλι

Σε ένα πραγματικό σύστημα OFDM, το λαμβανόμενο σήμα στο δέκτη έχει διέλθει μέσα από κανάλι που εισάγει κάποιου είδους παραμόρφωση (distortion) στο σήμα. Για τις ανάγκες της παρούσης αποτίμησης λαμβάνονται υπόψη δύο βασικοί τύποι πραγματικών καναλιών: *AWGN* και *Multipath*.

Στα *AWGN* κανάλια προστίθεται στο σήμα λευκός θόρυβος κατανομής Gauss. Το φάσμα του λευκού θορύβου είναι επίπεδο, εισάγοντας την ίδια ισχύ θορύβου σε όλα τα subcarrier και έτσι, το SNR ανά subcarrier είναι ίσο με το συνολικό SNR.

Στα *multipath* κανάλια, εκτός του *AWGN* θορύβου, υπάρχει πολυδιαδρομική διάδοση του σήματος, γεγονός που προκαλεί την εμφάνιση συχνο-επιλεκτικής συμπεριφοράς. Παρόλο που η ισχύς θορύβου είναι ίδια σε όλα τα subcarrier, ο σηματοθορυβικός λόγος (SNR) διαφέρει σε καθένα από αυτά, λόγω της συχνο-επιλεκτικότητας του ραδιοδιαύλου. Το συνολικό SNR του καναλιού είναι ο μέσος όρος των SNR ανά subcarrier. Τα κανάλια αυτά θα ονομάζονται για λόγους συντομίας ISI.

Για το κανάλι έχει δημιουργηθεί μοντέλο στο περιβάλλον MATLAB, που δέχεται ως είσοδο το σήμα όπως αυτό εξέρχεται από το μοντέλο του πομπού. Η επεξεργασία του σήματος γίνεται στη βασική ζώνη (base-band). Το πρώτο στάδιο είναι η συνέλιξη του σήματος με την κρουστική απόκριση του καναλιού, αν πρόκειται για ISI κανάλι. Στη συνέχεια, εφαρμόζεται μία σταθερή ολίσθηση συχνότητας και μία σταθερή καθυστέρηση στο χρόνο. Ακολουθεί η προσθήκη του *AWGN*. Μετά την εισαγωγή των παραμορφώσεων του καναλιού, το σήμα κλιμακώνεται με έναν σταθερό συντελεστή. Ο ρόλος του είναι να αποφεύγεται το clipping στην είσοδο του δέκτη και να διατηρεί την rms τιμή του σήματος στα ίδια επίπεδα με αυτά στην έξοδο του πομπού. Καθώς με την αλλαγή διαφόρων παραμέτρων του καναλιού ενδέχεται να παρουσιαστεί clipping, ο συντελεστής επιλέγεται μετά από δοκιμές για κάθε τύπο καναλιού. Οι βασικές παράμετροι του καναλιού είναι:

- i. το SNR
- ii. η ολίσθηση συχνότητας
- iii. η καθυστέρηση στο χρόνο
- iv. η κρουστική απόκριση του καναλιού
- v. ο συντελεστής κλιμάκωσης

Τα *multipath* κανάλια δημιουργούνται από εξειδικευμένα script. Βασικές παράμετροι για έναν τύπο *multipath* καναλιού είναι το λ και η μέγιστη διάρκεια του (T_h). Όσο μεγαλύτερη η τιμή του λ , τόσο πιο απότομα πέφτει η μέση ισχύς των path του καναλιού. Για κάθε επιλογή των παραμέτρων λ , T_h , δημιουργούνται πολλά στιγμιότυπα του καναλιού με τυχαίο τρόπο, ώστε το PDP⁴ τους να ακολουθεί την κατανομή

⁴Το *Power Delay Profile - PDP*, είναι η μέση ισχύς ενός καναλιού *multipath* συναρτήσει της καθυστέρησης. Ουσιαστικά, είναι η μέση ισχύς των ξεχωριστών path που αποτελούν την κρουστική απόκριση του καναλιού.

Rayleigh. Δηλαδή για έναν τύπο ISI καναλιού (επιλογή λ, T_h), δημιουργείται ένα σύνολο κρουστικών συναρτήσεων. Η αποτίμηση της απόδοσης του συστήματος για έναν τύπο καναλιού multipath, γίνεται μετά από μαζική προσομοίωση μέσα από το σύνολο των στιγμιότυπων του. Περισσότερα για τη μαζική προσομοίωση θα αναφερθούν παρακάτω.

5.2.2 Script Αποτίμησης (Evaluation Scripts)

Ο δέκτης καλείται να αποδιαμορφώσει ένα σήμα που έχει διέλθει από ένα κανάλι AWGN ή ISI με ολίσθηση χρόνου και συχνότητας. Για την αποτίμηση της επίδοσής του, δημιουργήθηκαν βοηθητικά script στο MATLAB (eval scripts). Ο σκοπός εδώ, είναι η σύγκριση των αποδιαμορφωμένων δεδομένων με αυτά που μεταδόθηκαν. Εκτός αυτού, ελέγχεται η έξοδος του *Time Synchronizer* (η χρονική ολίσθηση και η ολίσθηση συχνότητας) και η έξοδος του *Channel Estimator*. Γράφονται τα αντίστοιχα αρχεία διασταυρούμενου ελέγχου και τυπώνονται διαγνωστικά μηνύματα.

Τα χαρακτηριστικά μεγέθη που υπολογίζονται από τα eval scripts για κάθε προσομοίωση είναι :

Από την έξοδο του Symbol Demapper:

SER (Symbol Error Rate) Ανά τύπο διαμόρφωσης QAM και συνολικό. Αποτελεί ένα μέτρο συνολικής επίδοσης του τηλεπικοινωνιακού συστήματος. Τα σφάλματα ανίχνευσης μπορεί να προέρχονται από όλες τις πηγές θορύβου (εξω- και ενδο-συστημικές) που μεσολαβούν από την έξοδο του πομπού έως την έξοδο του δέκτη.

BER (Bit Error Rate) Ανά τύπο διαμόρφωσης QAM και συνολικό. Είναι παρόμοιο μέτρο με το SER, ωστόσο οι τιμές του ενδέχεται να είναι μικρότερες λόγω της ανοχής που προσφέρει η *κωδικοποίηση Gray*.

Από την έξοδο του Channel Estimator:

ISNR (Internal SNR) Ο σηματοθορυβικός λόγος, μετρημένος στην είσοδο του Symbol Demapper, ακριβώς πριν την αποδιαμόρφωση κατά QAM. Η συνολική ισχύς του θορύβου είναι μεγαλύτερη από αυτή της εισόδου του δέκτη, λόγω του εσωτερικού θορύβου που εισάγεται από τα διάφορα στάδια της διαδικασίας της αποδιαμόρφωσης.

INOISE (Internal Noise) Είναι το μέτρο του θορύβου που εισάγεται από τα διάφορα στάδια της αλυσίδας του δέκτη. Υπολογίζεται από το ISNR, έχοντας ως δεδομένο το SNR του καναλιού. Αποτελεί ένα μέτρο της επίδοσης του συστήματος του δέκτη μέχρι και τον Channel Estimator.

Από την έξοδο του Time Synchronizer:

SOF errors (Number of Start-of-Frame errors) (late, early, total, late ISI, early ISI, ISI total). Ο αριθμός των frame όπου η αρχή τους ανιχνεύτηκε λανθασμένα. Μετρώνται χωριστά τα SOF που ανιχνεύθηκαν νωρίτερα ή αργότερα από το σωστό σημείο συγχρονισμού. Εφόσον υπάρχει ανοχή στο σφάλμα συγχρονισμού, δεν προκαλούν όλα τα σφάλματα ISI.⁵ Μετρώνται ξεχωριστά τα ISI errors που προκύπτουν από frame που ανιχνεύθηκαν πρόωρα ή καθυστερημένα.

SOF error statistics (mean, mean abs, var). Τα βασικά στατιστικά μεγέθη για τα SOF errors. Το μέσο απόλυτο σφάλμα και η διακύμανσή, είναι τα κύρια μεγέθη με τα οποία αποτιμάται η επίδοση του αλγορίθμου του συγχρονισμού στο χρόνο.

Freq offset statistics (mean, mean abs, var). Τα βασικά στατιστικά μεγέθη για το σφάλμα ολίσθησης συχνότητας. Το μέσο απόλυτο σφάλμα και η διακύμανση, είναι τα κύρια μεγέθη για την αποτίμηση του αλγορίθμου συγχρονισμού στη συχνότητα.

⁵(ISI = Intersymbol Interference). Στις περιπτώσεις αυτές, το SOF error ξεπερνάει το κατώφλι ανοχής και το σφάλμα συγχρονισμού οδηγεί σε κατακόρυφη αύξηση του SER.

Για την απόκτηση κάποιου πλαισίου αναφοράς, τα παραπάνω μεγέθη υπολογίζονται τόσο για το μοντέλο του δέκτη στο MATLAB, όσο και για το κύκλωμα. Ουσιαστικά υπάρχουν δύο σύνολα από eval scripts για τα δύο είδη δεκτών. Η σύγκριση της επίδοσης του κυκλώματος με το μοντέλο, αποτελεί έναν δείκτη ποιότητας του κυκλώματος. Επιπλέον, από την αποτίμηση του μοντέλου, καθίστανται σαφή τα όρια των αλγορίθμων που έχουν επιλεγεί.

Κάνοντας χρήση των eval script για το δέκτη, επαληθεύτηκε η ορθή λειτουργία του σε κανάλια χωρίς και με AWGN, για διάφορες τιμές του SNR.

Μαζική Προσομοίωση (Sweep simulation)

Εφόσον στη διαδικασία της λήψης εμπλέκονται στοχαστικές διαδικασίες, θα πρέπει να εκτελεστεί ένας σχετικά μεγάλος αριθμός προσομοιώσεων για την απόκτηση ικανών αντιπροσωπευτικών μεγεθών για την αποτίμηση. Επιπλέον η ίδια η φύση των multipath καναλιών καθορίζει πως χρειάζονται πολλές προσομοιώσεις για ένα τύπο καναλιού. Εκτός αυτού, η αποτίμηση των διαφόρων μεγεθών που υπολογίζονται από τα eval scripts, έχει νόημα να γίνει συναρτήσει του SNR, καθώς είναι η πιο βασική παράμετρος που χαρακτηρίζει το κανάλι. Δηλαδή, το σύνολο των προσομοιώσεων πρέπει να επαναληφθεί για πολλές τιμές του SNR και τα αποτελέσματα να παρουσιαστούν με συγκεντρωτικό τρόπο.

Για το σκοπό αυτό δημιουργήθηκε μία σουίτα (suite) από MATLAB script για την υποστήριξη μαζικών προσομοιώσεων (sweep simulation). Ουσιαστικά πρόκειται για οργανωτικού τύπου script που χρησιμοποιούν τα script που περιγράφονται στις προηγούμενες ενότητες, για τις στοιχειώδεις λειτουργίες και εκτελούν της απαραίτητες διαχειριστικές εργασίες. Σημειώνεται πως, κατά το sweep simulation, παράγεται ένας μεγάλος όγκος δεδομένων.

Οι εργασίες που πρέπει να εκτελεστούν για την πραγματοποίηση του sweep simulation είναι:

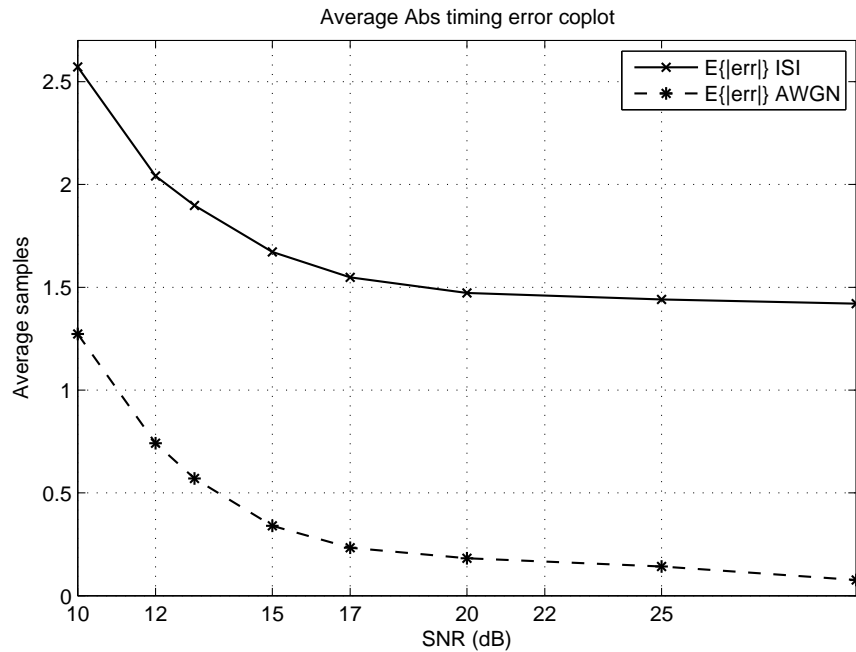
- i. Διαμόρφωση σήματος από το μοντέλο του πομπού. Η διάρκεια του σήματος πρέπει να είναι επαρκής για την αποτίμηση της στατιστικής συμπεριφοράς.
- ii. Μαζικό πέρασμα του διαμορφωμένου σήματος από διάφορα κανάλια με διαφορετικό SNR. Για κάθε SNR χρησιμοποιούνται πολλά στιγμιότυπα του ίδιου τύπου καναλιού. Ο αριθμός τους επιλέγεται έτσι ώστε να αποκτηθεί επαρκής γνώση για σωστή αναπαράσταση της στατιστικής συμπεριφοράς. Στα ISI κανάλια, για κάθε τύπο επιλέγεται και διαφορετική κρουστική απόκριση, από το σύνολο που χαρακτηρίζει το κανάλι.
- iii. Μαζική προσομοίωση του κυκλώματος (Modelsim) ή του μοντέλου του δέκτη (MATLAB).
- iv. Εκτέλεση των script αποτίμησης (sweep eval). Υπολογισμός στατιστικών μεγεθών και παραγωγή γραφημάτων.

5.2.3 Πορίσματα Αποτίμησης

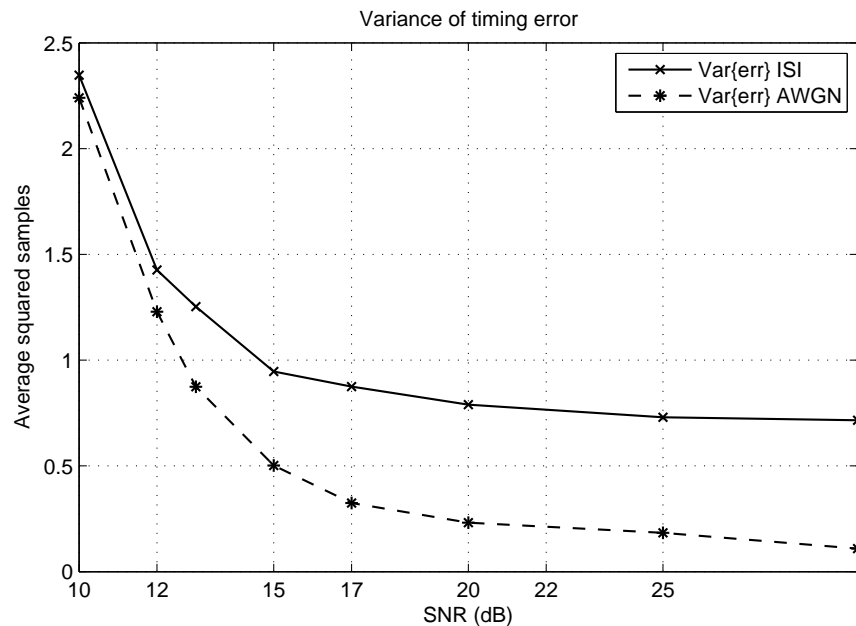
Σε όλες τις προσομοιώσεις διαμορφώθηκε μία τυχαία ακολουθία αποτελούμενη από 9 OFDM frame (63 data OFDM symbols). Σε όλα τα subcarrier επιλέχθηκε διαμόρφωση QAM-4. Προσομοιώθηκαν AWGN και ISI κανάλια. Επιλέχθηκαν οι εξής τιμές για το SNR [10, 12, 13, 15, 17, 20, 25, 30]. Στα AWGN δημιουργήθηκαν 600 κανάλια για κάθε SNR. Για τα ISI κανάλια επιλέχθηκε τύπος με $\lambda=1.0$ και $T_h = 16$. Δημιουργήθηκαν 800 στιγμιότυπα του καναλιού για κάθε SNR.

Παρακάτω παρουσιάζονται τα γραφήματα των μεγεθών που παράγονται από τα eval script και χαρακτηρίζουν τηλεπικοινωνιακά το σύστημα και, κατόπιν, γίνεται ανάλυση των αποτελεσμάτων.

Time Synchronization



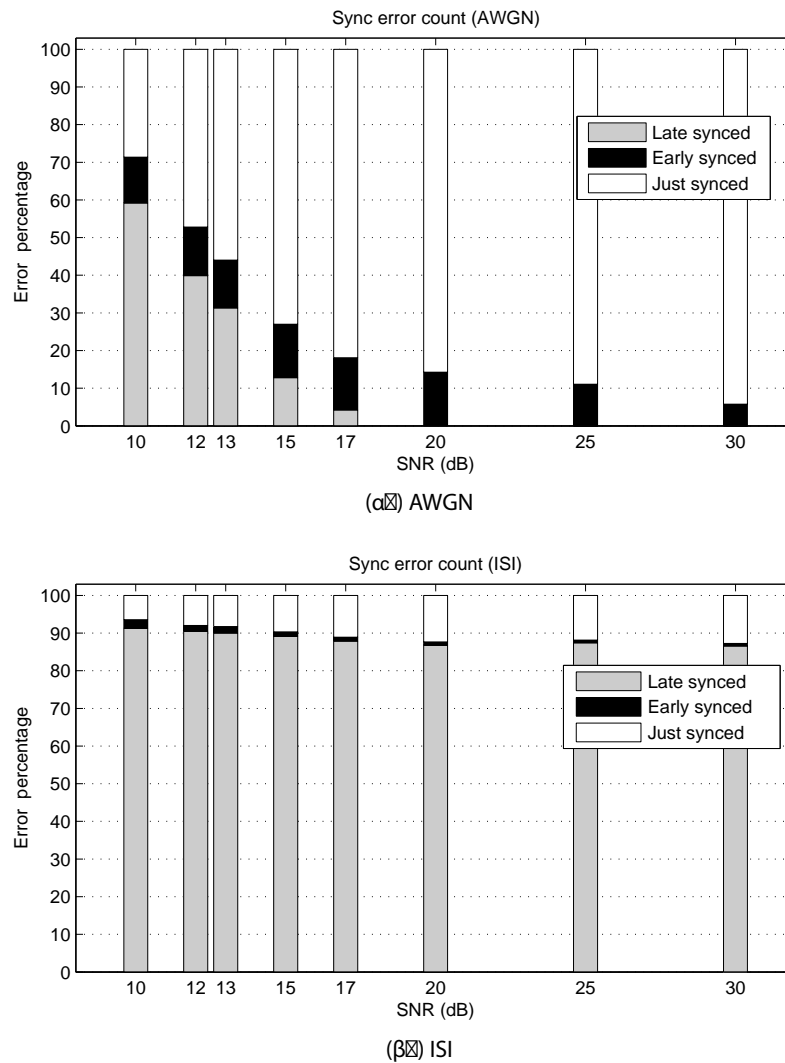
Σχήμα 5.1: Μέσο απόλυτο σφάλμα εκτίμησης SOF



Σχήμα 5.2: Διακύμανση σφάλματος εκτίμησης SOF

Στο Σχήμα 5.1 παρουσιάζεται το μέσο απόλυτο σφάλμα της εκτίμησης της αρχής του frame (SOF estimation), από κοινού για το AWGN και το ISI κανάλι. Στο Σχήμα 5.2 παρουσιάζεται η μεταβλητότητα αυτού του σφάλματος. Τα Σχήματα 5.3 α και β αποτελούν ιστογράμματα του πλήθους των σφαλμάτων συγχρονισμού διαχωρίζοντας τα πρόωρα από τα καθυστερημένα SOF. Σημειώνεται πως δεν υπάρχουν ISI σφάλματα.

Για το AWGN κανάλι και για $\text{SNR} > 12\text{dB}$, παρατηρείται ότι το μέσο απόλυτο σφάλμα είναι < 1.0 sample και η



Σχήμα 5.3: Ποσοστά σφαλμάτων συγχρονισμού

τυπική απόκλιση⁶ είναι < 1.2 samples. Καθώς τα σφάλματα για κάθε frame αποτελούν ανεξάρτητες τυχαίες μεταβλητές, μπορεί να εφαρμοστεί ο εμπειρικός στατιστικός κανόνας, σύμφωνα με τον οποίο το 99,7% των σφαλμάτων θα βρίσκεται εντός 3 τυπικών αποκλίσεων από τη μέση τιμή. Δηλαδή $err = 1.0 \pm 3.6$ samples, το οποίο είναι μικρότερο του διαστήματος ασφαλείας $S_{offset} = 9$.⁷ Καθώς η επίπτωση τέτοιου είδους σφαλμάτων μπορεί να αντισταθμιστεί πλήρως από την εκτίμηση καναλιού, θεωρούνται αμελητέα. Επίσης μετά τα 17dB το σφάλμα συγχρονισμού είναι πρακτικά μηδέν.

Η κατάσταση είναι διαφορετική στα ISI κανάλια. Εδώ για $SNR > 12$ dB ισχύει ότι το μέσο απόλυτο σφάλμα είναι < 2 samples και η τυπική απόκλιση < 1.2 samples. Σύμφωνα με τα παραπάνω μπορεί να υποθεθεί ότι το 99,7% των σφαλμάτων θα βρίσκεται εντός της περιοχής 2.0 ± 3.6 samples, που είναι μικρότερο του διαστήματος ασφαλείας $S_{offset} = 9$. Συνεπώς τα περισσότερα σφάλματα συγχρονισμού τόσο για το AWGN όσο και για το ISI κανάλι θεωρούνται αμελητέα, για $SNR > 12$ dB.

Στο ISI κανάλι παρατηρείται ένα κατώφλι σφάλματος στο 1.4 sample. Με τη βοήθεια του διαγράμματος (5.3α) φαίνεται ότι τα περισσότερα είναι καθυστερημένα SOF. Αυτό είναι λογικό δεδομένου ότι το multipath κανάλι εκτός από το απευθείας σήμα, παράγει και πολλαπλά καθυστερημένα αντίγραφα αυτού, προκαλώντας καθυστέρηση του μεγίστου της μετρικής. Ως μέτρο αντιστάθμισης, θα μπορούσε να αυξηθεί η τιμή του S_{offset} π.χ. κατά 2 samples. Αυτό θα μείωνε το μέσο απόλυτο σφάλμα στα ISI κανάλια,

⁶ Υπενθυμίζεται ότι η τυπική απόκλιση (standard deviation) είναι η τετραγωνική ρίζα της μεταβλητότητας (variance).

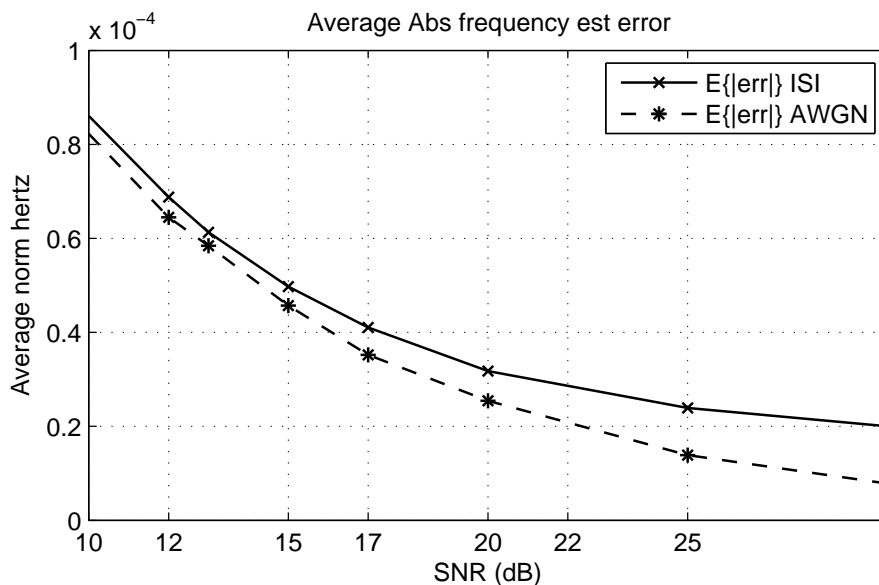
⁷Περισσότερα για το διάστημα ασφαλείας του SOF στην §2.2.3.

αλλά θα το αύξανε στα AWGN προκαλώντας πρώιμα σφάλματα συγχρονισμού. Χωρίς την αντιστάθμιση ωστόσο, δεν προκαλείται ISI από τα σφάλματα συγχρονισμού, καθώς το κατώφλι του 1.4 sample είναι μικρό σε σχέση με το περιθώριο σφάλματος που προσδίδεται από το S_{offset} . Αντιθέτως, στα AWGN κανάλια το μέσο απόλυτο σφάλμα τείνει στο 0 με την αύξηση του SNR και τα σφάλματα κατανομούνται τόσο σε πρώιμα όσο και σε καθυστερημένα (Σχήμα 5.3β).

Το ISI κανάλι παρουσιάζει κατώφλι στην μεταβλητότητα γύρω στα 0.7 squared samples, ενώ το AWGN όχι. Από τα παραπάνω είναι φανερή η επίδραση του multipath κατά τη λήψη. Φαίνεται επίσης η ανεκτικότητα που παρουσιάζει το σύστημα σε σφάλματα συγχρονισμού, ακόμα και για τιμές SNR της τάξης των 12dB.

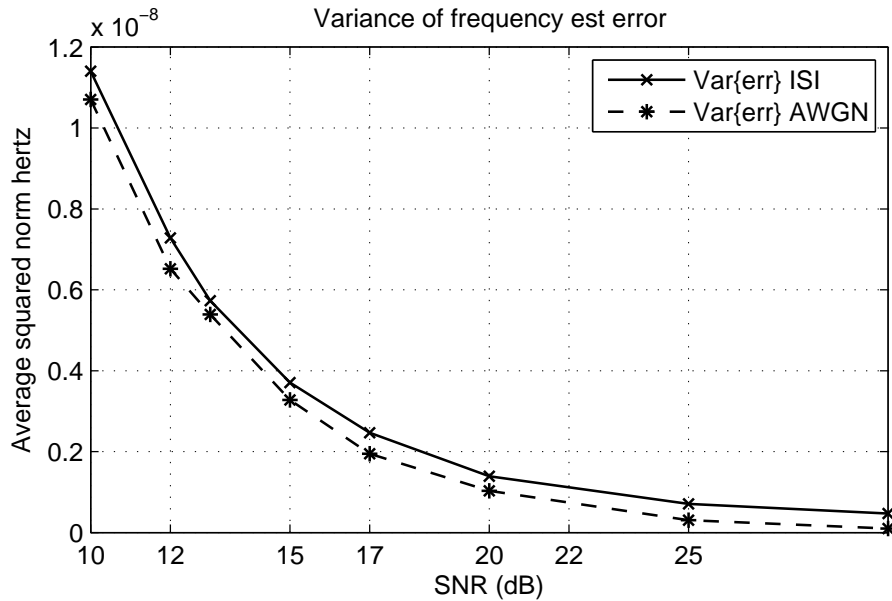
Frequency Synchronization

Εδώ, υπολογίζεται το μέσο απόλυτο σφάλμα εκτίμησης της ολίσθησης συχνότητας, καθώς και η διακύμανση του. Τα αποτελέσματα για το AWGN και το ISI κανάλι απεικονίζονται από κοινού στα Σχήματα 5.4 και 5.5. Οι μονάδες μέτρησης είναι κανονικοποιημένα Hertz.⁸ Για την απόκτηση ενός μέτρου σύγκρισης τα μεγέθη εκφράζονται και ως ποσοστό του *subcarrier spacing* (ssrc).⁹



Σχήμα 5.4: Μέσο απόλυτο σφάλμα εκτίμησης ολίσθησης συχνότητας

Το μέσο απόλυτο σφάλμα του AWGN βρίσκεται χαμηλότερα από το ISI, τουλάχιστον κατά 0.406×10^{-5} μονάδες. Και τα δύο μέσα σφάλματα προσεγγίζουν το μηδέν, με το AWGN να συγκλίνει ταχύτερα. Η μεταβλητότητα προσεγγίζει το μηδέν με την ίδια ταχύτητα και στα δύο είδη καναλιών με το ISI να βρίσκεται 0.429×10^{-9} μονάδες υψηλότερα από το AWGN. Το μέσο απόλυτο σφάλμα και για τις δύο περιπτώσεις είναι $< 6.877 \times 10^{-5}$ ή $< 0.88\%$ ssrc για $\text{SNR} > 12\text{dB}$ και η τυπική απόκλιση $< 8.53 \times 10^{-5}$. Χρησιμοποιώντας τον εμπειρικό στατιστικό κανόνα μπορεί να ειπωθεί ότι το 99.7% των σφαλμάτων συχνότητας θα βρίσκονται στο διάστημα $6.877 \pm 3 \times 8.53 \times 10^{-5}$, δηλαδή θα είναι $< 4.1\%$ ssrc για $\text{SNR} > 12\text{dB}$. Για $\text{SNR} > 17\text{dB}$, το ίδιο ποσοστό σφαλμάτων θα βρίσκεται στο $4.104 \pm 3 \times 4.964 \times 10^{-5}$, συνεπώς θα είναι $< 2.5\%$ ssrc.



Σχήμα 5.5: Διακύμανση σφάλματος εκτίμησης ολίσθησης συχνότητας

Channel Estimation

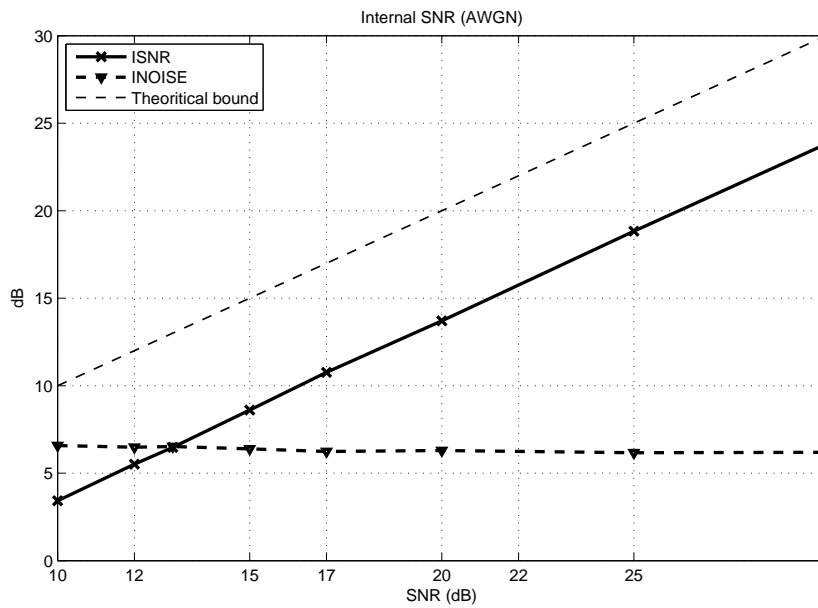
Εδώ, υπολογίζεται το ISNR του σήματος στην έξοδο του *Channel Estimator*. Παραγόμενο μέγεθος είναι το INOISE, που περιγράφει το μέγεθος του θορύβου που εισάγεται από τα υποσυστήματα που μεσολαβούν μέχρι και τον channel estimator. Όπως θα εξηγηθεί σε λίγο, η επικρατέστερη πηγή θορύβου είναι ο αλγόριθμος LS που υλοποιείται στον channel estimator. Εξάλλου από την μέχρι τώρα συζήτηση φάνηκε πως τα σφάλματα συγχρονισμού στο χρόνο δεν έχουν καμία επίδραση στο ISNR, ενώ τα σφάλματα συγχρονισμού συχνότητας αναμένεται να έχουν μικρή επίδραση. Στα Σχήματα 5.6, α και β, απεικονίζεται το ISNR και το INOISE συναρτήσει του εξωτερικού SNR για το AWGN και το ISI κανάλι. Στο διάγραμμα τοποθετείται και η μέγιστη τιμή του ISNR, που επιτυγχάνεται αν υποθεθεί πως δεν εισάγεται καθόλου θόρυβος στην πορεία του σήματος μέχρι τον channel estimator. Για να γίνει σαφέστερη η σύγκριση της επίδοσης μεταξύ AWGN και ISI, στο Σχήμα 5.7 παρουσιάζεται από κοινού το INOISE για τις δύο περιπτώσεις.

Στο AWGN κανάλι, το INOISE διατηρείται σχεδόν σταθερό, έχοντας μία ελαφρώς πτωτική τάση. Είναι μικρότερο από 6.4dB για SNR>12dB και μικρότερο από 6.1dB για SNR>25dB. Ενδεικτικές τιμές για το ISNR είναι 5.51dB (για SNR=12dB) και 18.83dB (για SNR=25dB). Παρατηρείται δηλαδή ένα κατώφλι θορύβου γύρω στα 6.1dB, που προέρχεται κατά κύριο λόγο από τον αλγόριθμο LS που έχει χρησιμοποιηθεί για την εκτίμηση του καναλιού. Το γεγονός αυτό επαληθεύτηκε με το μοντέλο MATLAB του δέκτη, θεωρώντας γνωστούς τους συντελεστές του καναλιού. Το αποτέλεσμα ήταν το ISNR να προσεγγίζει τη μέγιστη θεωρητική τιμή. Εξάλλου ο LS είναι ο πιο απλός αλγόριθμος εκτίμησης καναλιού, παρουσιάζοντας την χαμηλότερη απόδοση. Δεν εκμεταλλεύεται τη συσχέτιση που υπάρχει μεταξύ γειτονικών subcarrier, όπως για παράδειγμα ο MMSE.

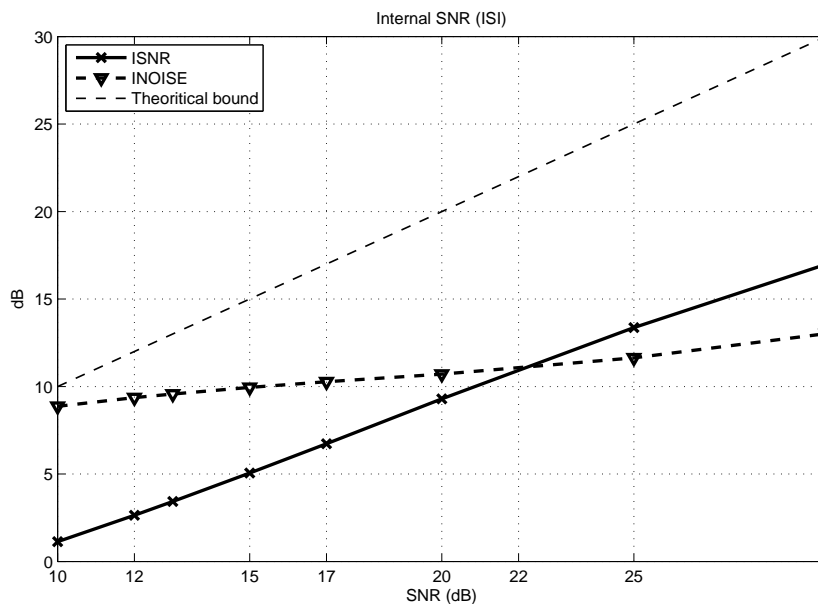
Στο ISI κανάλι, το INOISE ξεκινάει από τα 9.3dB για SNR=12dB και φτάνει στα 11.6dB για SNR=25dB, έχοντας αυξητική τάση. Τα αντίστοιχα μεγέθη για το ISNR είναι 2.63dB και 13.37dB. Μία πρώτη παρατήρηση είναι ότι οι τιμές του εσωτερικού θορύβου είναι υψηλότερες σε σχέση με τις αντίστοιχες για το AWGN κανάλι, όπως διακρίνεται καθαρότερα στο Σχήμα (5.7). Το φαινόμενο μπορεί να εξηγηθεί από τη συχνοεπιλεκτικότητα που παρουσιάζει το multipath κανάλι. Το φάσμα του καναλιού παρουσιάζει μεγάλες διακυμάνσεις στο πλάτος με αποτέλεσμα σε κάποια subcarrier το SNR να είναι υψηλό, ενώ σε άλλα να είναι

⁸ Για τον υπολογισμό της κανονικοποιημένης συχνότητας σε ένα ψηφιακό σύστημα, ισχύει ο τύπος $F = f/f_{sa}$, όπου f η συχνότητα και f_{sa} η συχνότητα δειγματοληψίας.

⁹ Το subcarrier spacing (η απόσταση μεταξύ δυο subcarrier), εκφρασμένο σε μονάδες κανονικοποιημένων Hertz είναι $\Delta_F = 1/N_o$. Για $N_o = 128 \Rightarrow \Delta_F = 0.0078$.



(α) AWGN

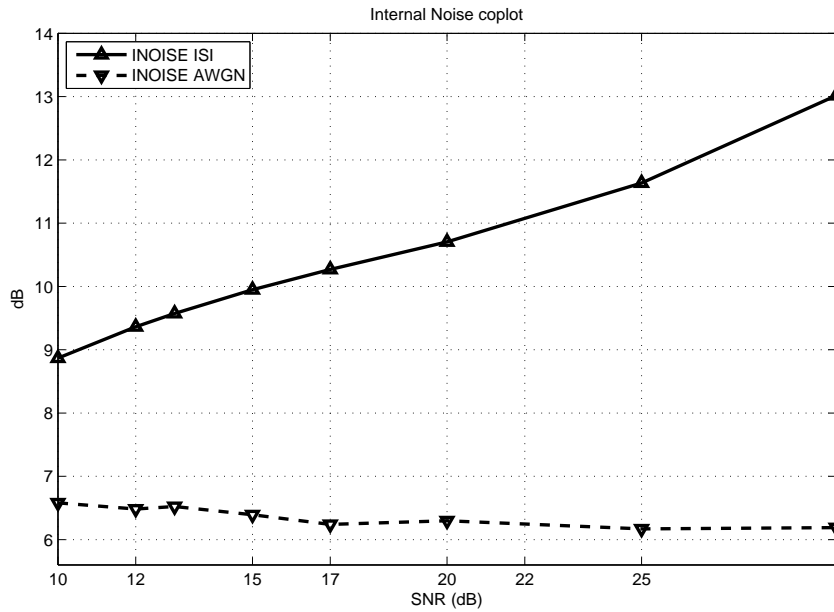


(β) ISI

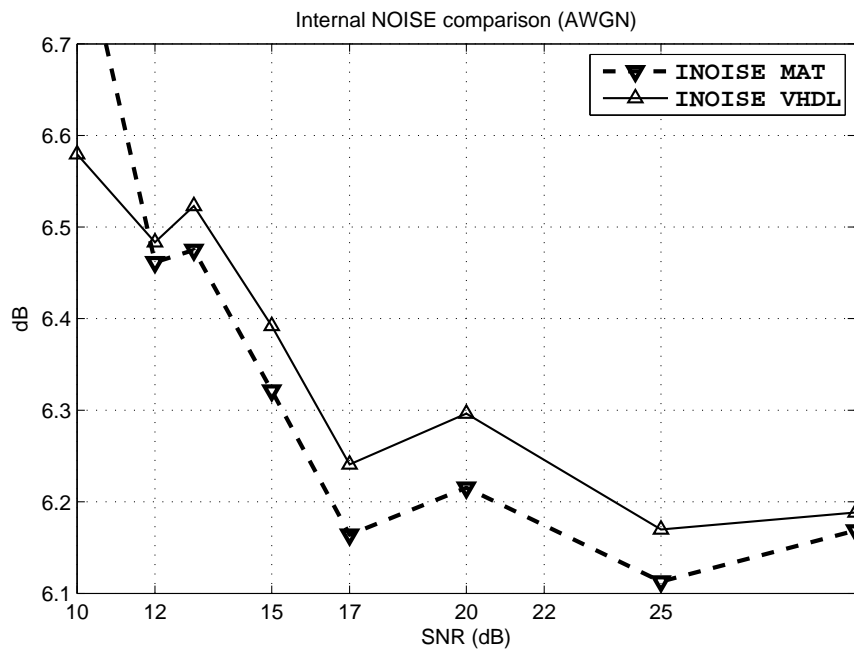
Σχήμα 5.6: ISNR και INOISE στην έξοδο του Channel Estimator

πολύ χαμηλό. Το εξωτερικό SNR, που χαρακτηρίζει το επίπεδο θορύβου στο κανάλι, αναφέρεται στο μέσο SNR των subcarrier. Όπως εξηγήθηκε στην προηγούμενη παράγραφο, η απόδοση του αλγορίθμου LS σε AWGN κανάλια, χειροτερεύει με την πτώση του SNR. Εφόσον το SNR ανά subcarrier μπορεί να φτάσει σε πολύ χαμηλές τιμές, η ποιότητα της εκτίμησης του καναλιού σε αυτά πέφτει ραγδαία, αυξάνοντας τον θόρυβο εκτίμησης. Το αποτέλεσμα είναι η αύξηση του μέσου θορύβου. Έτσι για την ίδια τιμή του εξωτερικού SNR, το ISI κανάλι παρουσιάζει υψηλότερο INOISE.

Μία άλλη παρατήρηση για το ISI κανάλι, είναι ότι η στάθμη θορύβου αυξάνεται με την αύξηση του SNR. Ενδεικτικά το INOISE παρουσιάζει αύξηση 2.27dB για SNR από 12dB στα 25dB. Και πάλι μπορεί να θεωρηθεί πως η αύξηση οφείλεται στον LS. Συγκρίνοντας το INOISE του ISI με αυτό του AWGN καναλιού παρατηρείται αύξηση 2.8dB (για 12dB SNR) και 5.4dB (για 25dB SNR).



Σχήμα 5.7: Σύγκριση INOISE μεταξύ AWGN και ISI καναλιού



Σχήμα 5.8: Σύγκριση INOISE μεταξύ κυκλώματος και μοντέλου

Ως ένα ακόμα σημείο αναφοράς παρουσιάζεται στο Σχήμα 5.8 ένα συγκριτικό γράφημα του INOISE για AWGN κανάλι, μεταξύ του κυκλώματος (VHDL) και του μοντέλου στο MATLAB (MAT). Το μοντέλο παρουσιάζει ελαφρώς μικρότερο θόρυβο, της τάξης των 0.08dB. Αυτή η διαφορά οφείλεται στην πεπερασμένη ακρίβεια των μαθηματικών πράξεων στο κύκλωμα, καθώς αυτό είναι το μόνο σημείο διαφοράς από το μοντέλο. Ωστόσο, πρόκειται για αμελητέα ποσότητα.

Symbol Error Rate • SER

Στο Σχήμα 5.9 παρουσιάζεται το γράφημα του ρυθμού σφαλμάτων συμβόλων (SER), από κοινού για το AWGN και το ISI κανάλι.

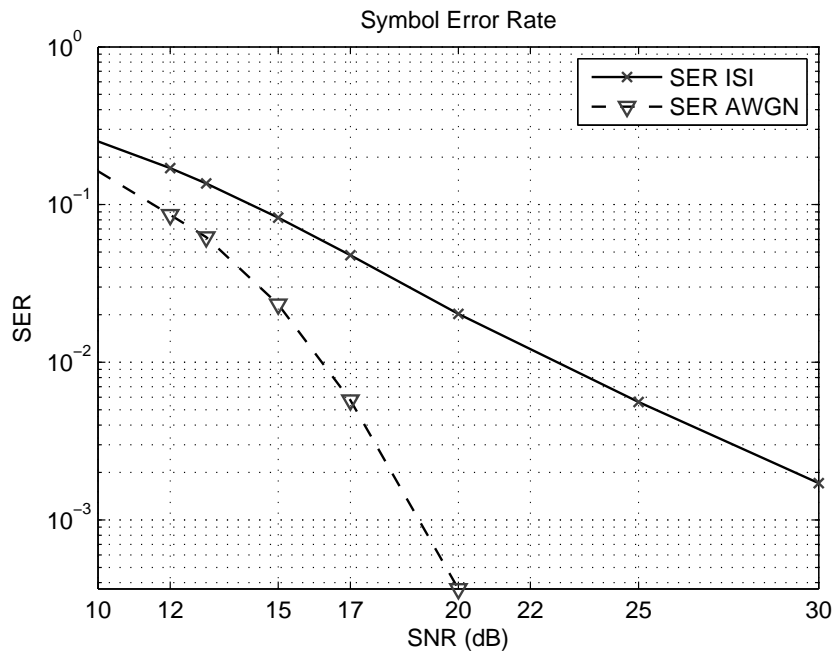
Στο AWGN κανάλι το SER πέφτει ραγδαίως ξεκινώντας από το 0.086 για SNR=12dB και φτάνοντας το 0.00037 για SNR=20dB. Για μεγαλύτερα SNR το SER είναι πρακτικά 0.¹⁰

Αντιθέτως στο ISI κανάλι το SER μειώνεται με βραδύτερο ρυθμό σε σχέση με το AWGN, ξεκινώντας από το 0.171 για SNR=12dB και φτάνοντας το 0.0056 για SNR=25dB. Δηλαδή το SER είναι από 2 έως 55 φορές μεγαλύτερο από το AWGN κανάλι.

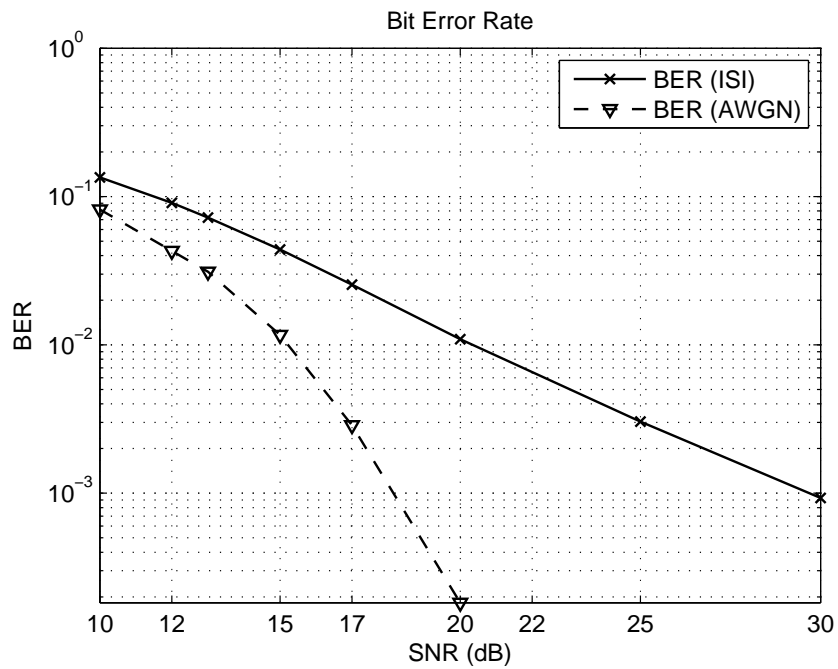
Bit Error Rate • BER

Τέλος στο Σχήμα 5.10 παρουσιάζεται το BER για τα δύο κανάλια. Η μορφή του είναι ίδια με αυτή του SER, καθώς χρησιμοποιείται QAM-4 διαμόρφωση. Αυτό πρακτικά σημαίνει πως, σχεδόν όλα τα εσφαλμένα υποσύμβολα ανιχνεύονται σε κάποια γειτονική περιοχή. Αυτό οδηγεί σε σφάλμα ενός bit. Γι' αυτό οι τιμές του BER είναι οι μισές σε σχέση με τις αντίστοιχες τιμές για το SER.

¹⁰ Στο γράφημα δεν παρουσιάζονται τιμές για μεγαλύτερους λόγους SNR, καθώς αυτό προϋποθέτει την εκτέλεση μεγαλύτερου αριθμού προσομοιώσεων. Ωστόσο τα δεδομένα που αποκτήθηκαν είναι αρκετά για τη σύγκριση με το ISI κανάλι.



Σχήμα 5.9: SER στην έξοδο του δέκτη



Σχήμα 5.10: BER στην έξοδο του δέκτη

Βιβλιογραφία

Βιβλία

- [Kan08] A. Κανάτας, Φ. Κωνσταντίνου και Γ. Πάντος. *Συστήματα Κινητών Επικοινωνιών*. Παπασωτηρίου, 2008.
- [LS06] Ye (Geoffrey) Li και Gordon Stüber, επιμελητές. *Orthogonal frequency division multiplexing for wireless communications*. Springer, 2006.
- [Pra04] Ramjee Prasad. *OFDM for Wireless Communications Systems*. Artech House, 2004.
- [PS02] John Proakis και Masoud Salehi. *Συστήματα Τηλεπικοινωνιών*. Επιμέλεια υπό Κ. Καρούμπαλος κ.ά. Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, 2002.
- [Rap96] Theodore S. Rappaport. *Wireless Communications, Principles & Practice*. Prentice Hall, 1996.
- [Pek03] Κ. Ζ. Πεκμεστζή. *Ψηφιακά Συστήματα VLSI*. Εθνικό Μετσόβειο Πολυτεχνείο, 2003.
- [BV01] Stephen Brown και Zvonko Vranesic. *Σχεδίαση Ψηφιακών Συστημάτων με τη Γλώσσα VHDL*. Τζιόλα, 2001.
- [Chu06] Pong P. Chu. *RTL Hardware Design Using VHDL*. Wiley-Interscience, 2006.
- [Mul06] Jean Michel Muller. *Elementary Functions - Algorithms and implementation*. 2nd edition. Birkhäuser, 2006.
- [Par00] Behrooz Parhami. *Computer Arithmetic, Algorithms and Hardware Designs*. Oxford University Press, 2000.
- [Ped07] Volnei A. Pedroni. *Σχεδιασμός Κυκλωμάτων με τη VHDL*. Κλειδάριθμος, 2007.
- [PM96] John G. Proakis και Dimitris G. Manolakis. *Digital Signal Processing, Principles, Algorithms and Applications*. 3rd edition. Prentice Hall, 1996.
- [WE96] N. H. Weste και K. Eshraghian. *Σχεδίαση Ολοκληρωμένων Κυκλωμάτων CMOS VLSI*. Παπασωτηρίου, 1996.
- [Fli04] Patrice Flichy. *Η Ιστορία της σύγχρονης επικοινωνίας*. Κάτοπτρο, 2004.

Άρθρα

- [Lee+06] Dong-U Lee κ.ά. "A Hardware Gaussian Noise Generator Using the Box-Muller Method and Its Error Analysis". Στο: *IEEE Transactions on Computers* 55.6 (2006).
- [MZB00] H. Minn, M. Zeng και V. K. Bargava. "On timing offset estimation for OFDM systems". Στο: *IEEE communications letters* 4.7 (2000).
- [Raj+06] Sreeraman Rajan κ.ά. "Efficient Approximations for the Arctangent Function". Στο: *IEEE Signal Processing Magazine* 111 (2006).

- [SC97] Timothy Schmidl και Donald Cox. "Robust frequency and timing synchronization for OFDM". Στο: *IEEE transactions on communications* 45.12 (1997).
- [Π01] Γ. Πάντος. *Θεωρία και Εφαρμογές OFDM*. Εθνικό Μετσόβειο Πολυτεχνείο. 2001.
- [SM06] Yushi Shen και Ed Martinez. *Channel Estimation in OFDM Systems*. Freescale semiconductor. 2006.
- [Tan91] Ping Tak Peter Tang. *Table-Lookup Algorithms for Elementary Functions and Their Error Analysis*. Argonne National Laboratory, Mathematics and Computer Science Division. 1991.

Διάφορα

- [Φ06] Ε. Φάσσα και Β. Χουσέα. "Μελέτη & προσομοίωση συστήματος OFDM". Μεταπτυχιακή διπλ. εργασ. Εθνικό Μετσόβειο Πολυτεχνείο, 2006.
- [Ava] *Avalon Interface Specifications*. 1.2. Altera. 2009.
- [Ofd] *Implementing OFDM Modulation for Wireless Communications*. 1.0. Altera. 2008.