Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων
Εργαστηριο Μικροϋπολογιστων και Ψηφιακων Συστηματων

Digital processing system for Coherent Detection Wind LiDAR with SoC FPGA

Διπλωματικη Εργασια

του

Γεωργίου-Ελευθερίου Λιάτσου

Επιβλέπων: Δημήτριος Σούντρης
Καθηγητής Ε.Μ.Π.

Εργαστηριο Μικροϋπολογιστων και Ψηφιακων Συστηματων
Αθήνα, Μάρτιος 2020

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Μικροϋπολογιστών και Ψηφιακών Συστημάτων

Digital processing system for Coherent Detection Wind LiDAR with SoC FPGA

Διπλωματικη Εργασια

του

Γεωργίου-Ελευθερίου Λιάτσου

Επιβλέπων: Δημήτριος Σούντρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 6 Μαρτίου 2020.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Δημήτριος Σούντρης	Παναγιώτης Τσανάκας	Αλέξανδρος Παπαγιάννης
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π

Αθήνα, Μάρτιος 2020

(Υπογραφή)

.....

Γεωργιος-Ελευθεριος Λιατσος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2020 – All rights reserved

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Μικροϋπολογιστών και Ψηφιακών Συστημάτων

Copyright ©–All rights reserved Γεώργιος-Ελευθέριος Λιάτσος, 2020.
Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, 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 τον συγγραφέα.

Περίληψη

Το LiDAR είναι μία τεχνική ενεργούς, οπτικής τηλεπισκόπησης, με πληθώρα τομέων εφαρμογής, συμπεριλαμβανομένων της γεωγραφίας, της ρομποτικής και της αυτόματης πλοήγησης οχημάτων. Από τις διάφορες παραλλαγές LiDAR, κάθε μία εκ των οποίων στοχέυει σε διαφορετικό ζητούμενο και έχει διαφορετικούς τομείς εφαρμογής, ιδιαίτερο ενδιαφέρον παρουσιάζουν τα Coherent Doppler LiDAR. Η λειτουργία τους στηρίζεται στο φαινόμενο Doppler και στοχεύει στην ανίχνευση της ταχύτητας του ανέμου. Η παρούσα διπλωματική εργασία αφορά τη σχεδίαση συστήματος ψηφιακής επεξεργασίας των δεδομένων του Wind LiDAR σε SoC FPGA, που οδηγεί στον προσδιορισμό της ταχύτητας του ανέμου. Περιλαμβάνει ακόμα τη μελέτη των παραμέτρων που επηρεάζουν τη διαδικασία, στην προσπάθεια βελτιστοποίησής της. Στα πλαίσια της συστημικής προσέγγισης, πέραν της ψηφιακής επεξεργασίας, σκοπός είναι το ενσωματωμένο σύστημα να εκτελεί τον ελεγχο της περιστορφής του LiDAR, καθώς και να είναι ικανό να προσδιορίζει τη διεύθυνση από την οποία λαμβάνονται οι μετρήσεις του.

Λέξεις Κλειδιά

SoC FPGA, Wind LiDAR, Doppler, ανάλυση συχνότητας, βελτιστοποίηση παραμέτρων

Abstract

The LiDAR method is a type of active, optic remote sensing, with a majority of application fields, such as geography, robotics and navigation of autonomous vehicles. Of the various LiDAR systems developed, aiming at different targets and therefore having different fields of application, the Coherent Doppler LiDAR is of great interest. Its operation is based on the Doppler Effect and aims at the detection of wind velocity. This diploma thesis concerns the design of a digital system to process Wind LiDAR data, in order to determine wind velocity. It also includes an exploration of the variables affecting the procedure, in order to optimize it. In the boundaries of a systemic approach, except for the digital processing, the embedded system is also set to control the LiDAR's rotation, as well as to determine the direction at which the measurements are taken.

Keywords

SoC FPGA, Wind LiDAR, Doppler, frequency analysis, variables optimization

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντά μου, Καθηγητή του ΕΜΠ Δημήτριο Σούντρη, ο οποίος μου έδωσε την ευκαιρία να εκπονήσω τη διπλωματική μου εργασία στο Εργαστήριο Μικροϋπολογιστών και Ψηφιακών Συστημάτων (Microlab) στο ΕΜΠ.

Θα ήθελα, ακόμη, να ευχαριστήσω τους μεταδιδακτορικούς ερευνητές Νικόλαο Ζομπάκη και Γεώργιο Λεντάρη, καθώς και τον υποψήφιο διδάκτορα Βασίλειο Λέοντα για την καθοδήγηση και τη βοήθεια που μου προσέφεραν για την εκπόνηση της διπλωματικής μου. Η συνεχής τριβή και οι συζητήσεις μας με βοήθησαν να αντιληφθώ γρηγορότερα ζητήματα που αφορούσαν την εργασία μου, ενώ σε κώδικα του Γεωργίου Λεντάρη βασίστηκε μέρος των πειραμάτων που πραγματοποιήθηκαν. Ευχαριστώ επίσης όλα τα μέλη του Microlab για το ευχάριστο περιβάλλον εργασίας και τη βοήθειά τους, όποτε αυτή ζητήθηκε.

Ένα μεγάλο ευχαριστώ σε όλη την οικογένεια και του φίλους μου, που με στηρίζουν καθ΄ όλη τη διάρκεια της ζωής και των σπουδών μου και μου δίνουν κίνητρο για να συνεχίζω να κυνηγάω τους στόχους μου. Ιδιαίτερες ευχαριστίες στον θείο μου Θοδωρή, ο οποίος με βοήθησε με τη διόρθωση του αγγλικού κειμένου, και στην κοπέλα μου Μαρία, η οποία με βοήθησε με τη διόρθωση του ελληνικού κειμένου και με στήριξε σε όλη τη διάρκεια της διπλωματικής, δίνοντάς μου αυτοπεποίθηση σε στιγμές αγχώδεις και πιεστικές.

Contents

Περίληψη 1

Abstract 3

Ευχαριστίες 5

Contents 7

List of Figures 9

List of Tables 11

Εκτεταμένη Περίληψη 13

1 Introduction 37
- 1.1 LiDAR and Applications . . . 37
- 1.2 Problem Statement . . . 38
- 1.3 Thesis Scope . . . 39

2 Wind LiDAR 41
- 2.1 Wind LiDAR Fundamentals . . . 41
- 2.2 Doppler Effect . . . 41
- 2.3 Coherent Detection . . . 42
- 2.4 Frequency Domain Analysis . . . 46
- 2.5 Challenge: Signal Analysis with extremely low SNR . . . 47
- 2.6 Related Research and Implementations . . . 48

3 SoC FPGA 51
- 3.1 FPGA Fundamentals . . . 51
- 3.2 FPGA Architecture . . . 52
  - 3.2.1 Configurable Logic Blocks . . . 52
  - 3.2.2 Hard Blocks . . . 52
- 3.3 Benefits and Applications . . . 53
- 3.4 System on a Chip . . . 54

3.5 FPGA in LiDAR Applications 55

4 Developed Processing System **57**
4.1 Proposed System Architecture 57
4.2 Noise Compensation Method: Averaging 59
4.3 Proposed System Implementation 63
4.3.1 Frequency Analysis and Averaging 63
4.3.2 LiDAR Direction Determination 67
4.4 Zynq Ultrascale 102+ MPSoC 68

5 Experimental Results **71**
5.1 Optimum Variables Exploration 71
5.1.1 Gate length effect on computation accuracy 73
5.1.2 Computing error in relation to the number of accumulations, for different noise conditions 74
5.2 VHDL Design Verification 84
5.2.1 $L_g = 32$ Design 84
5.2.2 $L_g = 64$ Design 85
5.2.3 Utilization and Timing 86

6 Conclusions and Future Work **89**

References **93**

List of Figures

1 Η διάταξη ενός CDWL [27] . . . 16
2 Φάσμα συχνοτήτων σήματος εισόδου [23] . . . 18
3 Προτεινόμενο Σύστημα . . . 21
4 Φάσμα συχνοτήτων για ένα μόνο παράθυρο οπισθοσκεδασμένης ακτινοβολίας. 22
5 Αντιμετώπιση θορύβου με συσσώρευση φασμάτων . . . 23
6 Αλυσίδα επεξεργασίας δεδομένων . . . 26
7 Κατανομή δειγμάτων εισόδου στους επεξεργαστικούς πόρους . . . 26
8 Σήματα του κινητήρα για την περιστροφή στην (I) ωρολογιακή και (II) αντιωρολογιακή φορά . . . 27
9 Zynq Ultrascale 102+ MPSoC [33] . . . 29
10 $f_{error} = f(L_g)$, για P = 512 accumulations . . . 32
11 $f_{error} = f(L_g)$, για P = 2048 accumulations . . . 32
12 $f_{error} = f(P, f_{var})$ με $L_g = 32$ . . . 33
13 $f_{error} = f(P, f_{var})$ με $L_g = 64$ . . . 33

2.1 The CDWL layout [27] . . . 42
2.2 ADC operation [5] . . . 43
2.3 CDWL operating principle [15] . . . 44
2.4 Power Spectral Density of signal R(F) [15] . . . 46
2.5 Low noise input frequency spectrum [23] . . . 47
2.6 Mean average (μ) and standard deviation (σ) of sunlight noise during day and night [29] . . . 50

3.1 FPGA parallel to CPU . . . 51
3.2 The flip-flop and LUT symbols . . . 52
3.3 CPU + FPGA vs SoC FPGA [7] . . . 54

4.1 Proposed Design . . . 57
4.2 Azimuth and Altitude [2] . . . 58
4.3 Frequency detection without (left) and with (right) parabola fitting . . . 59
4.4 Power Spectrum for a single backscattered pulse . . . 60
4.5 Noise elimination through spectral accumulation . . . 61
4.6 Feuilleté Model [24] . . . 62

4.7 Data processing chain 64
4.8 Input data routing to processing units 65
4.9 PSD mapping to system's memory resources 65
4.10 Motors' (I) clock- and (II) counter-clockwise rotation indicating signals . . . 67
4.11 Zynq Ultrascale 102+ MPSoC [33] 69

5.1 $f_{error} = f(L_g)$ for P = 512 accumulations 73
5.2 $f_{error} = f(L_g)$ for P = 2048 accumulations 74
5.3 $f_{error} = f(P)$ with $L_g = 32$ 75
5.4 $f_{error} = f(P)$ with $L_g = 32$ 76
5.5 $f_{error} = f(P)$ with $L_g = 32$ 77
5.6 $f_{error} = f(P)$ with $L_g = 32$ 78
5.7 $f_{error} = f(P)$ with $L_g = 64$ 79
5.8 $f_{error} = f(P)$ with $L_g = 64$ 80
5.9 $f_{error} = f(P)$ with $L_g = 64$ 81
5.10 $f_{error} = f(P)$ with $L_g = 64$ 82
5.11 $f_{error} = f(P, f_{var})$ with $L_g = 32$ 83
5.12 $f_{error} = f(P, f_{var})$ with $L_g = 64$ 83
5.13 Spectral Accumulation L_g=32 84
5.14 Spectral Accumulation L_g=32 (close up) 84
5.15 Single Gate Spectrum L_g=32 85
5.16 Spectral Accumulation L_g=64 86
5.17 Single Gate Spectrum L_g=64 86

List of Tables

1 Πίνακας αληθείας για κύκλωμα μετρητή γωνίας 28
2 Χαρακτηριστικά ZCU102 29
3 Απαιτούμενος αριθμός μετρήσεων για την επίτευξη ακρίβειας 0.1 MHz 34
4 Χρήση Πόρων 34
5 Χρονική Επίδοση 34

4.1 FFT IP Configuration 66
4.2 Counter Value Change Truth Table 68
4.3 ZCU102 features 69

5.1 Accumulations P required to achieve 0.1MHz accuracy 83
5.2 Resources Utilization 86
5.3 Timing Performance 87
5.4 Angle Detection Implementation Utilization 87

Εκτεταμένη Περίληψη

1. Εισαγωγή

LiDAR και εφαρμογές

Η τεχνική LiDAR (Light Detection And Ranging) είναι ένα είδος ενεργούς, οπτικής τηλεπισκόπησης. Η αρχή της ενεργούς τηλεπισκόπησης είναι η εξής: μια πηγή εκπέμπει ακτινοβολία στην ατμόσφαιρα και ένας δέκτης συγκεντρώνει την οπισθοσκεδαζόμενη από τα διάφορα σώματα της ατμόσφαιρας ακτινοβολία προς ανάλυση. Η LiDAR τεχνική είναι οπτική, αφού σε αντίθεση με άλλες τεχνικές ενεργούς τηλεπισκόπησης (SONAR, SODAR, RADAR) χρησιμοποιεί ακτινοβολία σε εύρος οπτικών συχνοτήτων (φωτόνια).

Το εύρος των μηκών κύματος laser που μπορούν να χρησιμοποιηθούν, σε συνδυασμό με τη δυνατότητα ακριβούς επιλογής συχνότητας laser οδήγησαν στη δημιουργία πολλών ειδών LiDAR συστημάτων. Τέτοια είναι το LiDAR συντονισμού φθορισμού (resonance fluorescence) , το LiDAR διαφορικής απορρόφησης (differential absorption LiDAR), το Raman LiDAR, τα Coherent Doppler και τα LiDAR εντοπισμού κατεύθυνσης (direction detection) , καθώς και τα LiDAR λευκού φωτός (white light) και οπτικού χτενίσματος (optical comb).

Χρησιμοποιούνται σήμερα συνήθως για δημιουργία χαρτών υψηλής ανάλυσης, με εφαρμογή σε τομείς όπως η αρχαιολογία, η γεωδαισία, η γεωγραφία, η γεωλογία, η γεωμορφολογία, η σεισμολογία, η δασοκομία, η ρομποτική, η φυσική της ατμόσφαιρας, η καθοδήγηση με laser, η αυτόματη πλοήγηση οχημάτων κ.α.

Η παρούσα διπλωματική εργασία ασχολείται με τα Coherent Detection Doppler LiDAR. Το σύστημα αυτό εντοπίζει τη διαφορά της συχνότητας μεταξύ της εκπεμπόμενης ακτινοβολίας και της οπισθοσκεδαζόμενης από τα σωματίδια της ατμόσφαιρας, λόγω της κίνησης αυτών. Αξιοποιώντας το φαινόμενο Doppler, το σύστημα προσδιορίζει την ταχύτητα της κίνησης των σωματιδίων και κατ επέκταση του ανέμου. Η ακρίβεια μέτρησης και το μεγάλο εύρος ανίχνευσης που προσφέρουν αυτά τα συστήματα οδηγεί στη χρήση τους σε εύρος εφαρμογών.

Επιστημονικοί οργανισμοί, όπως η NASA με το HARLIE LiDAR κωνικής ανίχνευσης και η ESA με το LiDAR της αποστολής ADM-Aeolus, τα χρησιμοποιούν για κάθετες μετρήσεις προφίλ ανέμου. Σχετικές εφαρμογές αφορούν και την φυσική και συγκεκριμένα την αστρονομία. Πέραν της έρευνας, τα Wind LiDAR χρησιμοποιούνται και για τη βελτιστοποίηση της απόδοσης αιολικών πάρκων. Οι μετρήσεις του ανέμου μπορούν να αυξήσουν την παραγωγή ενέργειας, να δώσουν σημαντικές πληροφορίες για την προστασία του εξοπλισμού, αλλά και

να χρησιμοποιηθούν για την επιλογή κατάλληλης τοποθεσίας για τη δημιουργία νέων εγκαταστάσεων. Εκτός αυτών, τα Coherent Doppler LiDAR έχουν εφαρμογή και στον τομέα της μετεωρολογίας, ενώ έχουν χρησιμοποιηθεί μέχρι και στους Ολυμπιακούς Αγώνες του 2008 στο Πεκίνο, για μετρήσεις ανέμων στο επίπεδο της θάλασσας για το άθλημα της ιστιοπλοΐας.

Ζητήματα

Το πρόβλημα με την επεξεργασία μετρήσεων των Coherent Detection Doppler LiDAR είναι διττό, αφορώντας τόσο το μεγάλο φόρτο Ψηφιακής Επεξεργασίας Σήματος, όσο και την ανάγκη για αποθήκευση μεγάλου όγκου δεδομένων. Ένώ τελικά είναι εφικτό να προσδιοριστει η ταχύτητα του ανέμου με μεγάλη ακρίβεια, δεν αρκεί μία μόνο μέτρηση. Αυτό συμβαίνει διότι εξωγενείς και ενδογενείς παράγοντες του συστήματός τους, οδηγούν το θόρυβο στις μετρήσεις τους σε πολύ χαμηλά επίπεδα SNR (Signal to Noise Ratio), καθιστώντας τις ιδιαίτερα αναξιόπιστες. Η πιο συνήθης μέθοδος αντιμετώπισης του προβλήματος, είναι η χρήση του μέσου όρου ενός πλήθους μετρήσεων. Αυτό, ωστόσο, προϋποθέτει την αποθήκευση και επεξεργασία μεγάλου όγκου δεδομένων.

Η παραπάνω λύση οδηγεί με τη σειρά της σε μια νέα πρόκληση, από τη σκοπιά του hardware, αυτή της αποθήκευσης των δεδομένων. Δεδομένου του ρυθμού με τον οποίο μπορούν να παρέχουν δεδομένα οι σύγχρονοι ADC (Analog to Digital Converter) των LiDAR, η χρήση κατάλληλου σκληρού δίσκου είναι σημαντική. Και ενώ η χρήση μέσων αποθήκευσης παλαιότερης γενιάς ενδεχομένως να μη μπορεί να υποστηρίξει αυτές τις απαιτήσεις, ούτε η χρηση SSD σύγχρονης τεχνολογίας είναι η βέλτιστη λύση. Ο λόγος είναι η αύξηση του κόστους και της πολυπλοκότητας του ελέγχου της σχεδίασης, αλλά και το ίδιο το περιεχόμενο των δεδομένων που λαμβάνονται. Η αποθήκευση χιλιάδων μετρήσεων που περιέχουν υψηλά επίπεδα θορύβου και δε μπορούν να προσφέρουν παρά ένα μόνο αποτέλεσμα μέτρησης, δεν αξιοποιεί τη σύγχρονη αυτή τεχνολογία στο μέγιστο βαθμό.

Όλοι οι λόγοι που αναφέρθηκαν θα πρέπει να ληφθούν υπ όψιν κατά τη σχεδίαση του συστήματος επεξεργασίας και η βελτιστοποίηση μπορεί να προέλθει μόνο μέσα από τον πειραματισμό και την έρευνα.

Το αντικείμενο της διπλωματικής εργασίας

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η πρόταση ψηφιακού συστήματος που θα εκτελεί την επεξεργασία δεδομένων από Doppler Wind LiDAR για τον προσδιορισμό της ταχύτητας του ανέμου και θα ελέγχει τη θέση του τηλεσκοπίου, άρα και τη διεύθυνση λήψης των μετρήσεων.

Περιέχει ακόμα αποτελέσματα πειραμάτων στην κατεύθυνση της βελτιστοποίησης της μεθόδου αντιμετώπισης του θορύβου (averaging). Σε πρώτο στάδιο, η μελέτη αυτή αφορά τη σχέση μεταξύ της χωρικής και χρονικής ανάλυσης του συστήματος με την ακρίβεια μέτρησης, ενώ έπειτα ασχολείται με το συμβιβασμό μεταξύ της υψηλής ακρίβειας και του χαμηλού αριθμού μετρήσεων που είναι ζητούμενα.

Με σκοπό τη βελτιστοποίηση των πόρων, θα γίνει χρήση SoC FPGA για την υλοποίηση του συστήματος. Εκτός των δυνατοτήτων προσαρμογής στην εφαρμογή, παραλληλίας και pipelined λειτουργίας που προσφέρει, η χρήση του μπορεί να αντιμετωπίσει και άλλα προαναφερθέντα ζητήματα. Αρχικά, μπορεί να οδηγήσει στην ανάπτυξη αποδοτικού, γρήγορου και ακριβούς συστήματος επεξεργασίας σε πραγματικό χρόνο στην πλατφόρμα FPGA, καταργώντας την ανάγκη χρήσης επιπλέον αποθηκευτικού χώρου για post-processing. Μειώνει επίσης τον φόρτο του επεξεργαστή, αναλαμβάνοντας τους βαρείς και χρονοβόρους υπολογισμούς, αλλά και το κόστος και την κατανάλωση του συστήματος. Επιπλέον, η χρήση σύγχρονης SoC πλατφόρμας καθιστά το σύστημα μικρό σε μέγεθος, φορητό και εύκολα μετακινήσιμο και προσαρμόσιμο σε οποιοδήποτε σύστημα Wind LiDAR. Ο επεξεργαστής ARM και τα περιφερειακά της πλατφόρμας μπορούν να χρησιμοποιηθούν για την επικοινωνία με το σύστημα LiDAR και την ανάπτυξη απομακρυσμένης, φιλικής προς το χρήστη επικοινωνίας.

2. Wind LiDAR

Βασικές Αρχές

Ιδιαίτερα ενδιαφέρουσα είναι η κατηγορία των Wind LiDAR. Η δυνατότητα ακριβούς μέτρησης των ανέμων και της θερμοκρασίας της ατμόσφαιρας είναι σημαντική για τομείς όπως τα μοντέλα πρόβλεψης του καιρού, η πολιτική και στρατιωτική αεροπορία και γενικότερα οι μεταφορές στην ατμόσφαιρα, η αμυντική βιομηχανία, η έρευνα σχετικά με τη δυναμική της ατμόσφαιρας και η παρακολούθηση της κλιματικής αλλαγής. Τα Wind LiDAR χωρίζονται σε δύο κατηγορίες, ανάλογα με την τεχνική ανίχνευσης του ανέμου. Αυτά που ανήκουν στην πρώτη κατηγορία χρησιμοποιούν την άμεση ανίχνευση κίνησης (Direct Motion Detection) εκμεταλλευόμενα τον ορισμό της ταχύτητας:

$$\vec{V} = \frac{\vec{dr}}{dt}$$

Στην περίπτωση αυτή, είναι απαραίτητη η ύπαρξη ιχνηλατών (tracers), όπως σύννεφα ή αερολύματα, για την μέτρηση της μεταβολής της θέσης τους, είτε μέσω εικόνων είτε μέσω LiDAR, με το πέρασμα του χρόνου.

Στη δεύτερη κατηγορία, ανήκουν τα LiDAR που χρησιμοποιούν το φαινόμενο Doppler (Doppler Shift) και η ταχύτητα του ανέμου υπολογίζεται σύμφωνα με τον τύπο:

$$\Delta\vec{w} = -2k\vec{V}$$

όπου Δw η μεταβολή της συχνότητας, k ο κυματαριθμός του laser και V η ταχύτητα του ανέμου.

Φαινόμενο Doppler

Το φαινόμενο Doppler αφορά την παρατηρούμενη αλλαγή στη συχνότητα ενός κύματος από παρατηρητή που βρίσκεται σε σχετική κίνηση με την πηγή των κυμάτων. Πήρε το όνομά

του από τον Αυστριακό φυσικό Christian Doppler που το πρότεινε το 1842 για ακουστικά κύματα. Έχει όμως εφαρμογή και στα ηλεκτρομαγνητικά κύματα και ακτινοβολία.

Στη γενική μορφή του φαινομένου [20], για κύματα που διαδίδονται σε υλικό, η σχέση μεταξύ παρατηρούμενης συχνότητας (f') και εκπεμπόμενης (πραγματικής) συχνότητας (f) δίνεται από τον τύπο :

$$f' = (\frac{u \pm u_0}{u \mp u_s})f,$$

όπου

- u η ταχύτητα διάδοσης του κύματος
- u_0 η ταχύτητα του παρατηρητή ως προς το μέσο διάδοσης
- u_s η ταχύτητα της πηγής ως προς το μέσο διάδοσης

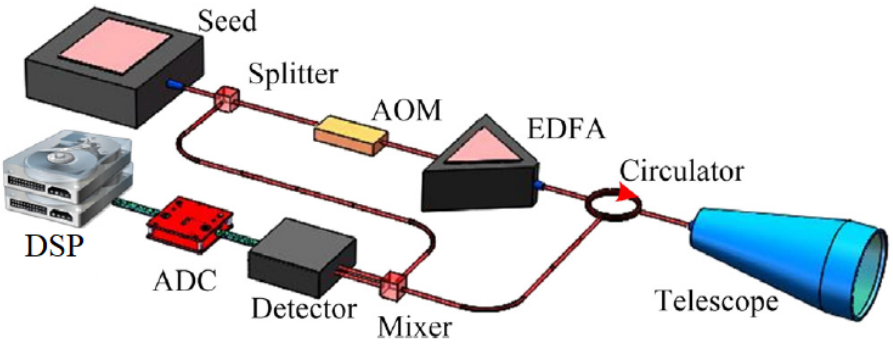
Υπολογίζοντας τη μεταβολή της συχνότητας της ακτινοβολίας και αξιοποιώντας τον παραπάνω τύπο προκύπτει η ζητούμενη (ακτινική) ταχύτητα του ανέμου. Ο καθορισμός της μεταβολής της συχνότητας με LiDAR συστήματα μπορεί να γίνει είτε άμεσα (Direct Detection Doppler LiDAR), είτε "συνεκτικά" (Coherent Detection Doppler LiDAR).

Στα Direct Detection Doppler LiDAR χρησιμοποιείται ένας διευκρινιστής (discriminator) οπτικής συχνότητας ή ένας αναλυτής φάσματος, για τη μετατροπή της μεταβολής της συχνότητας λόγω Doppler είτε σε μεταβολή οπτικής έντασης ή ισχύος είτε σε τοπική κατανομή έντασης ή ισχύος, τα οποία έπειτα ανιχνεύονται άμεσα.

Στα Coherent Detection Doppler LiDAR, από την άλλη πλευρά, το οπισθοσκεδαζόμενο σήμα αναμειγνύεται με laser σε έναν τοπικό ταλαντωτή και καταλήγει σε σήμα με συχνότητα ίση (πέραν κάποιου δεδομένου offset) με τη μεταβολή Doppler, λόγω της κίνησης των σωματιδίων του ανέμου.

Συνεκτική Ανίχνευση

Πιο συγκεκριμένα, η διάταξη που χρησιμοποιείται σε ένα Coherent Detection Wind LiDAR είναι η εξής:



Σχήμα 1: Η διάταξη ενός CDWL [27]

Στο πρώτο στάδιο μία πηγή παράγει laser συνεχούς κύματος (Continous Wave). Ένας splitter χρησιμοποιείται για να διαχωρίσει την ακτίνα σε δύο τμήματα, το ένα εκτών οποίων είναι προς εκπομπή, ενώ το άλλο θα μεταβεί σε τοπικό ταλαντωτή για να χρησιμοποιηθεί στην πορεία. Το συνεχές laser προς εκπομπή διέρχεται από ακουστικό-οπτικό μετατροπέα (Acoustic-Optical Modulator), ο οποίος το χωρίζει σε διαδοχικούς παλμούς συχνότητας 80MHz και διαρκείας 300ns. Το laser ενισχύεται διερχόμενο από έναν οπτικό ενισχυτή. Ο ενισχυτής αυτός ονομάζεται Erbium Doped Fiber Amplifier. Ο ενισχυμένος παλμός διέρχεται στη συνέχεια από κυκλοφορητή, η δέσμη συγκεντρώνεται και εκπέμπεται στην ατμόσφαιρα μέσω του τηλεσκοπίου που έπεται στη διάταξη του Σχήματος 1. Το ίδιο τηλεσκόπιο είναι υπεύθυνο για τη συλλογή των οπισθοσκεδαζόμενων φωτονίων, τα οποία αναμειγνύονται με την ακτίνα που είχε μεταβεί προηγουμένως στον τοπικό ταλαντωτή, με τη χρήση ειδικού μίκτη 3dB. Το παραγόμενο σήμα μετράται από έναν ισορροπημένο ανιχνευτή και ψηφιοποιείται με τη χρήση ADC (Analog to Digital Converter). Τα ψηφιακά δεδομένα είναι πλέον έτοιμα προς επεξεργασία, είτε σε πραγματικό χρόνο είτε αποθηκεύονται με σκοπό να επεξεργαστούν και να μελετηθούν μετέπειτα.

Ανάλυση στον τομέα της συχνότητας

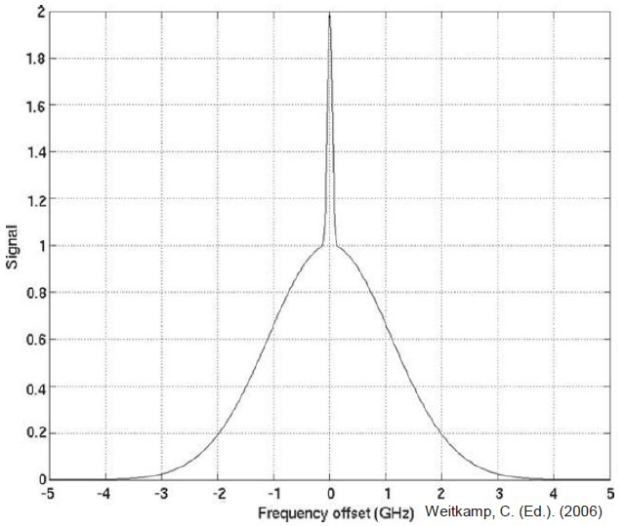
Αμελώντας σε πρώτο στάδιο το θόρυβο, το φάσμα συχνοτήτων της οπισθοσκεδαζόμενης ακτινοβολίας που ανιχνεύεται με το LiDAR αποτελείται από δύο κύριες συνιστώσες (Σχήμα 2). Η πρώτη οφείλεται στην σκέδαση Mie και αφορά τη σκέδαση του φωτός από αερολύματα στα χαμηλότερα 4.5 χιλιόμετρα της ατμόσφαιρας, περιοχή που περιέχεται στην ακτίνα ανίχνευσης. Εκεί η εκπεμπόμενη ακτινοβολία μπορεί να συναντήσει σφαιρικά σωματίδια με διαμέτρους προσεγγιστικά ίδιες με το μήκος κύματός της. Η δεύτερη συνιστώσα οφείλεται στη μοριακή σκέδαση ή αλλιώς σκέδαση Rayleigh, που είναι αποτέλεσμα της ελαστικής σκέδασης της ακτινοβολίας από μόρια με αισθητά μικρότερες διαμέτρους από το μήκος κύματός της. Η ένταση του οπισθοσκεδαζόμενου σήματος σε αυτή την περίπτωση είναι αντιστρόφως ανάλογη του μήκους κύματος της ακτινοβολίας υψωμένου εις την τετάρτη.

Η συνιστώσα του Σχήματος 2 με τη χαμηλότερη ένταση και τη μεγαλύτερη διακύμανση είναι η συνιστώσα Rayleigh. Ακολουθεί τη γκαουσιανή κατανομή, με μέση τιμή την κεντρική συχνότητα του λαμβανόμενου σήματος. Η συνιστώσα Mie έχει μεγαλύτερη ένταση και μικρότερη διακύμανση από την μοριακή συνιστώσα, με κέντρο τη ζητούμενη συχνότητα.

Πηγές Θορύβου

Η εύρεση της συχνότητας που αντιστοιχεί στο μέγιστο του φάσματος είναι σύνθετη, διότι σε πραγματικές συνθήκες περιέχει υψηλού επιπέδου θόρυβο. Πρακτικά, για μια τέτοια διάταξη και μέτρηση μπορεί να χρειαστεί να αντιμετωπιστούν επίπεδα SNR (Signal to Noise Ratio) έως και -20dB.

Πρώτο και κύριο πρόβλημα είναι η χαμηλή ένταση της οπισθοσκεδαζόμενης ακτινοβολίας. Είναι δεδομένο ότι, τόσο η μοριακή σκέδαση, όσο και η σκέδαση από αερολύματα στην ατμόσφαιρα είναι ισοτροπικές. Αυτό σημαίνει πως και στις δύο περιπτώσεις επιτρέπεται η



Σχήμα 2: Φάσμα συχνοτήτων σήματος εισόδου [23]

σκέδαση των φωτονίων της δέσμης φωτός προς οποιαδήποτε κατεύθυνση του τρισδιάστατου χώρου, μειώνοντας αισθητά τον αριθμό φωτονίων που επιστρέφουν σε αντίθετη κατεύθυνση και ανιχνεύονται από το τηλεσκόπιο. Η μείωση αυτή, ποιοτικά μεταφράζεται σε χαμηλής έντασης χρήσιμη πληροφορία στο σήμα που ανιχνεύεται, η οποία εξαρτάται και από το μήκος κύματος του laser που εκπέμπει το LiDAR.

Άλλη σημαντική πηγή θορύβου για την ακτινοβολία που λαμβάνεται είναι η ηλιακή ακτινοβολία. Η παραπλανητική συμβολή της οπισθοσκέδασης της ηλιακής ακτινοβολίας στις μετρήσεις του Wind LiDAR κατά τη διάρκεια της ημέρας είναι τεράστια, εξαιτίας της μεγάλης της έντασης και της ποικιλίας των συνιστωσών της σε μήκη κύματος. Ο θόρυβος που οφείλεται στην ηλιακή ακτινοβολία μπορεί να είναι ακόμα και κατά τάξεις μεγαλύτερος από το ασθενές σήμα επιστροφής του LiDAR, ιδίως όταν έχει οπισθοσκεδαστεί από στερεές γήινες επιφάνειες ή πάγο, ή στην περίπτωση του Wind LiDAR από πυκνά σύννεφα.

Εκτός των παραπάνω εξωτερικών πηγών θορύβου για τις μετρήσεις, όπως κάθε μηχανικό σύστημα, έτσι και σε αυτή την περίπτωσή, το ίδιο το LiDAR εισάγει θόρυβο στις μετρήσεις του. Καταρχάς, υπάρχει ο shot noise του ταλαντωτή του συστήματος, θόρυβος που ακολουθεί προσεγγιστικά την κατανομή Poisson. Εκτός αυτού, υπάρχει και ο θερμικός θόρυβος που εμφανίζεται σε οποιονδήποτε ηλεκτρικό αγωγό, συνεπώς και σε όλους τους αγωγούς του συστήματός που μελετάται.

3. SoC FPGA

Βασικές Αρχές FPGA

Ένα field-programmable gate array (FPGA) είναι ένα ημιαγώγιμο ολοκληρωμένο κύκλωμα, σχεδιασμένο να διαμορφώνεται από το χρήστη του μετά την κατασκευή του. Η διαμόρφωση αυτή πραγματοποιείται με τη χρήση μιας γλώσσας περιγραφής υλικού (HDL). Τα FPGA

αποτελούνται από έναν πίνακα προγραμματίσιμων λογικών block και μια ιεραρχία επαναδιαμορφώσιμων συνδέσεων μεταξύ τους. Στα block αυτά μπορούν να υλοποιηθούν από απλές λογικές πύλες έως σύνθετα κυκλώματα, ενώ συχνή είναι η ύπαρξη και στοιχείων μνήμης, είτε σε μορφή απλών flip-flop, είτε πιο ολοκληρωμένων στοιχείων. Σε αντίθεση με τους επεξεργαστές, στα FPGA κάθε ξεχωριστό επεξεργαστικό έργο ανατίθεται σε διαφορετικούς πόρους και έτσι δεν υπάρχει το φαινόμενο του ανταγωνισμού. Αποτέλεσμα είναι η αυτονομία στην απόδοση των τμημάτων υπολογισμού μιας εφαρμογής, ανεξαρτήτως του υπολογιστικού φόρτου στο υπόλοιπο FPGA, ανά πάσα στιγμή.

Αρχιτεκτονική FPGA

Τα FPGA αποτελούνται από πεπερασμένο αριθμό πόρων με προγραμματίσιμες διασυνδέσεις μεταξύ τους για τη σχεδίαση ψηφιακών κυκλωμάτων και πόρους εισόδου/εξόδου για την επικοινωνία του κυκλώματος με το περιβάλλον του. Όλοι αυτοί οι πόροι χωρίζονται σε Configurable Logic Blocks και Hard Blocks.

- Configurable Logic Blocks: Σε αυτή την κατηγορία ανήκουν τα flip-flops και οι LUTs. Τα flip-flops είναι δυαδικοί καταχωρητές μετατόπισης που συγχρονίζουν τη λογική του κυκλώματος, αποκτώντας μία τιμή σε κάθε ακμή ρολογιού και διατηρώντας την μέχρι την επόμενη. Τα Look Up Tables είναι ψηφιακά στοιχεία που υλοποιούν πίνακες αληθείας, με προκαθορισμένη τιμή εξόδου για κάθε διαφορετική τιμή εισόδου.

- Hard Blocks: Σε αυτή την κατηγορία ανήκουν οι πολλαπλασιαστές, τα DSP blocks και διάφορα ενσωματωμένα στοιχεία μνήμης. Οι πολλαπλασιαστές είναι προσχεδιασμένα κυκλώματα αποδοτικής εκτέλεσης της σύνθετης πράξης του πολλαπλασιασμού, για διευκόλυνση των Logic Blocks. Τα DSP blocks μπορούν να εκτελέσουν ακόμα πιο σύνθετες και υπολογιστικά απαιτητικές πράξεις, ενώ τα ενσωματωμένα στοιχεία μνήμης (πχ Block RAM) χρησιμοποιούνται για την απαλλαγή των Logic Blocks από το φόρτο της αποθήκευσης μεγάλων πινάκων δεδομένων.

Πλεονεκτήματα και Εφαρμογές

Τα πλεονεκτήματα που μπορούν να προσφέρουν τα FPGA συμπεριλαμβάνουν:

- **Επιτάχυνση**: Δυνατότητα εξατομικευμένης προσαρμογής του υλικού σε κάθε εφαρμογή, με χρήση παραλληλίας και διοχέτευσης για μείωση του χρόνου εκτέλεσης.

- **Ευελιξία**: Δυνατότητα μεταβολής της αρχιτεκτονικής με κάθε επανέναρξη της συσκευής και ενσωμάτωσης αλλαγών με τη φόρτωση νέων παραλλαγών.

- **Ενσωμάτωση**: Δυνατότητα ενσωμάτωσης σε chip με επεξεργαστές, γρήγορους διαύλους εισόδου-εξόδου, στοιχεία μνήμης, μηχανές DSP και άλλα, με αποτέλεσμα τη μείωση του κόστους επικοινωνίας μεταξύ τους.

- **Κόστος ιδιοκτησίας**: Αποφυγή των εφάπαξ (non-recurring) εξόδων των ASICs και υποστήριξη μεγάλου χρόνου ζωής.

Τα προαναφερθέντα πλεονεκτήματα οδηγούν στη χρήση FPGA σε μεγάλο εύρος εφαρμογών, συμπεριλαμβανομένων ηλεκτρικών συσκευών, επιστημονικών αλλά και ευρείας κατανάλωσης, ενσύρματων και ασύρματων επικοινωνιών, επίγειων και εναέριων μεταφορών, ιατρικών και οικονομικών. Πολύ σημαντική είναι η συμβολή τους στον τομέα του Computing υψηλής απόδοσης, σε servers, υπερυπολογιστές, συστήματα RADAR, αλλά και στη σχεδίαση ολοκληρωμένων κυκλωμάτων, όντας το τέλειο αντικείμενο για σχεδίαση πρωτότυπων ASIC και hardware υπολογιστών.

System on a Chip

Τα SoC FPGA (System on Chip FPGA) είναι σύνθετα ολοκληρωμένα κυκλώματα που περιέχουν ψηφιακά και αναλογικά στοιχεία, καθώς και στοιχεία που λειτουργούν σε ραδιοσυχνότητες, όλα σε ένα chip. Η ενσωμάτωση όλων των λειτουργιών σε ένα υπόστρωμα οδηγεί σε μείωση της κατανάλωσης, του κόστους και του μεγέθους, συγκριτικά με σχεδιάσεις που χρησιμοποιούν διαφορετικά chip. Η χρήση επεξεργαστή και FPGA στην ίδια πλακέτα επιτρέπει επίσης μεγαλύτερο εύρος ζώνης για την επικοινωνία μεταξύ τους και καλύτερη συνεργασία μεταξύ hardware και software για την εκτέλεση υπολογισμών. Για τους λόγους αυτούς, τα SoC FPGA είναι ιδιαίτερα διαδεδομένα στους τομείς των ενσωματωμένων συστημάτων και του Internet of Things.

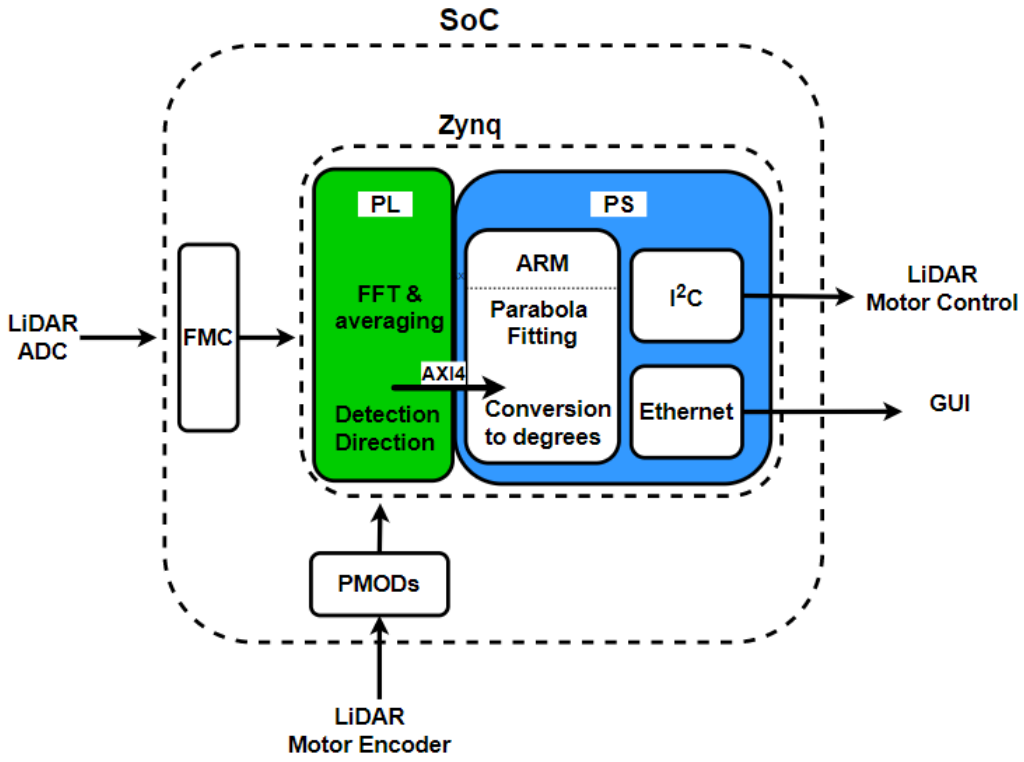
Τα σημερινά SoC FPGA βασίζονται ως επί το πλείστον σε ARM επεξεργαστές και παράγονται από τη Xilinx (Zynq 7000 AP SoC, Zynq UltraScale+ MPSoC), την Intel (Stratix 10 SoC, Arria 10 SoC, Arria V SoC, Cyclone V SoC) και τη Microsemi (SmartFusion SoC, SmartFusion 2 SoC).

4. Ανάπτυξη προτεινόμενου συστήματος

Αρχιτεκτονική Προτεινόμενου Συστήματος

Από το Κεφάλαιο 2 φαίνεται πως οι μετρήσεις που λαμβάνονται από LiDAR σύστημα μπορούν να είναι ιδιαίτερα θορυβώδεις, ιδίως για μακρινές αποστάσεις και για "δυσμενείς" ατμοσφαιρικές συνθήκες. Σε ορισμένες περιπτώσεις μάλιστα, αναμένονται SNR σε επίπεδα των -20dB, γεγονός που καθιστά το σήμα που λαμβάνεται για ένα παράθυρο εκπεμπόμενης ακτινοβολίας αναξιόπιστο. Η απαίτηση για υψηλή ακρίβεια στην τελική μέτρηση για προσδιορισμό της ταχύτητας του ανέμου, γεννά την ανάγκη καταπολέμησης του θορύβου.

Εκτός της επεξεργασίας σήματος, μία συστημική προσέγγιση οδηγεί είτε στην ανάπτυξη περισσότερων κυκλωμάτων είτε στην αξιοποίηση των υπολοίπων πόρων του SoC για τον έλεγχο της κατεύθυνσης του LiDAR από όπου λαμβάνονται οι μετρήσεις, αλλά και την αλληλεπίδραση με το χρήστη.



Σχήμα 3: Προτεινόμενο Σύστημα

Το προτεινόμενο σύστημα, φαίνεται στο Σχήμα 3. Ο ADC που χρησιμοποιείται θεωρείται πως παρέχει δεδομένα με το ίδιο ρυθμό με τον οποίο το FPGA τα επεξεργάζεται. Στο FPGA θα υλοποιηθεί η δύσκολη υπολογιστική εργασία της μετατροπής του σήματος εισόδου στον τομέα της συχνότητας με τον Fast Fourier Transform και της συσσώρευσης των αποτελεσμάτων για τη λήψη ενός μέσου όρου των μετρήσεων. Στο PL θα υλοποιηθεί, επίσης, κύκλωμα υπολογισμού της διεύθυνσης του LiDAR από δεδομένα που παρέχει ο κωδικοποιητής των κινητήρων του. Το σύστημα αυτό θα επικοινωνεί και με τον επεξεργαστή του SoC, στον οποίο θα εκτελείται και η επεξεργασία του μέσου όρου των φασμάτων που υπολογίζεται στο PL, για τον καθορισμό της ταχύτητας του ανέμου. Τέλος, το SoC μπορεί να επικοινωνεί μέσω της IIC θύρας του με τους κινητήρες περιστροφής του LiDAR για τον έλεγχο της θέσης του και να αξιοποιήσει το Ethernet για ανάπτυξη γραφικού περιβάλλοντος (GUI) απομακρυσμένης επικοινωνίας με το χρήστη.

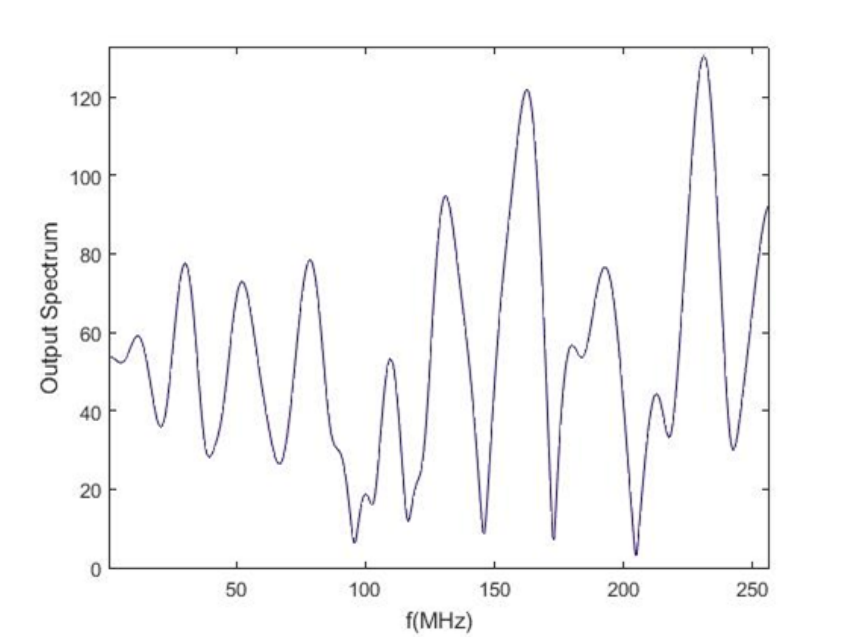
Στα πλαίσια αυτής της εργασίας αναπτύχθηκαν και δοκιμάστηκαν τα δύο κύρια τμήματα αυτής της σχεδίασης στο PL, το κύκλωμα που υλοποιεί τη μετατροπή στον τομέα της συχνότητας και υπολογίζει το μέσο όρο των φασμάτων που προκύπτουν και το κύκλωμα προσδιορισμού της κατεύθυνσης του τηλεσκοπίου του LiDAR.

Αντιμετώπιση θορύβου: μέσος όρος μετρήσεων

Για την αντιμετώπιση του θορύβου θα πρέπει να μελετηθεί το σήμα που αναμένεται να ληφθεί (Σχήμα 2), προσθέτοντας τώρα πια, πέραν των δύο γνωστών συνιστωσών του φάσμα-

τος (Mie και Rayleigh) θόρυβο. Για την μοντελοποίησή του θα αξιοποιηθεί το γεγονός ότι τα σήματα που προκύπτουν από συνεκτική ανίχνευση (coherent detection) είθισται να μοντελοποιούνται ως ζωνοπερατά rf σήματα με Πρόσθετο Λευκό Γκαουσιανό Θόρυβο (AWGN). Αυτό το μοντέλου θορύβου ακολουθεί την κανονική κατανομή, καθιστώντας όλα τα διακριτά σημεία του ασυσχέτιστα μεταξύ τους και τυχαία κατανεμημένα.

Η προαναφερθείσα στοχαστικότητα είναι αυτή που θα χρησιμοποιηθεί για την καταπολέμηση του θορύβου των μετρήσεων του συστήματος. Υλοποιώντας το μοντέλο σε MATLAB και λαμβάνοντας το φάσμα συχνοτήτων για ένα παράθυρο του σήματος, θεωρώντας ένταση AWGN ίση με -20dB, προκύπτει η θορυβώδης έξοδος του Σχήματος 4. Συγκεκριμένα, για αυτό το παράδειγμα η κεντρική συχνότητα της ακτινοβολίας είναι τα 80MHz, συχνότητα που είναι αδύνατον να εντοπιστεί από αυτό το θορυβώδες φάσμα συχνοτήτων.



Σχήμα 4: Φάσμα συχνοτήτων για ένα μόνο παράθυρο οπισθοσκεδασμένης ακτινοβολίας.

Η έξοδος του συστήματος στο Σχήμα 4 απέχει πολύ από το ιδανικό φάσμα του Σχήματος 2. Δεδομένου ότι ο θόρυβος είναι ομοιόμορφα κατανεμημένος και τυχαίος, επαναλαμβάνοντας τη διαδικασία πολλές φορές και αθροίζοντας τα επιμέρους φάσματα είναι δυνατόν να λυθεί το πρόβλημα. Πράγματι, όλα τα επιμέρους θορυβώδη σήματα αποτελούνται από μία "καθαρή" κορυφή στη συχνότητα προς ανίχνευση και από πολλές "θορυβώδεις" κορυφές, οι οποίες κατανέμονται τυχαία και με τυχαίο μέτρο σε διάφορες άλλες συχνότητες του φάσματος. Αθροίζοντας, λοιπόν, αρκετά φάσματα, τα οποία έχουν όλα την κοινή κορυφή στην συχνότητα προς ανίχνευση και διαφορετικές κορυφές στις υπόλοιπες "θορυβώδεις" συχνότητες, αρχίζει να προκύπτει ένα φάσμα στο οποίο ο WGN αποτελεί απλώς ένα offset, πάνω από το οποίο προκύπτει η εικόνα του επιθυμητού-ιδανικού φάσματος, όπως φαίνεται και στο Σχήμα 5. Είναι πλέον εφικτό να προσδιοριστεί η συχνότητα του οπισθοσκεδαζόμενου σήματος και κατ επέκταση η διαφορά της από τη συχνότητα του απεσταλμένου σήματος, οπότε και η ταχύτητα

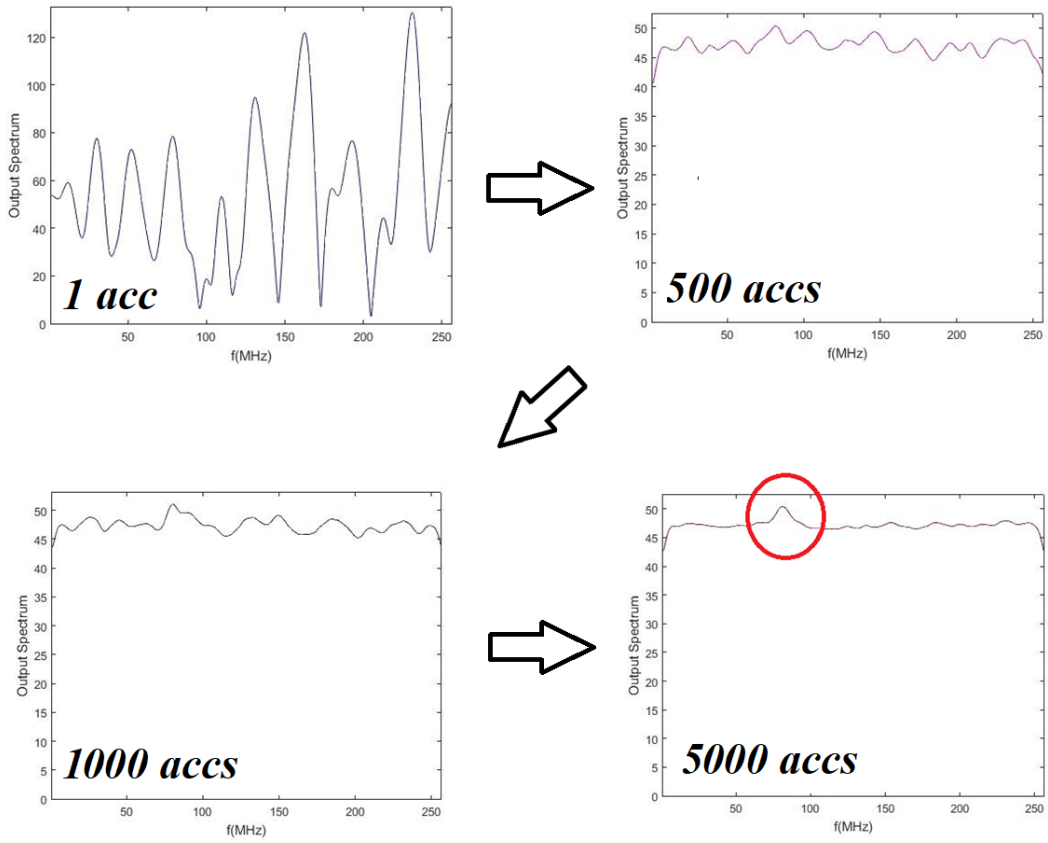
του ανέμου.

Το σύστημα του Wind LiDAR που χρησιμοποιείται είναι pulsed, δεν εκπέμπει δηλαδή συνεχές laser, αλλά σήμα χωρισμένο σε παλμούς. Η χωρική ανάλυση του LiDAR δίνεται από τον τύπο:

$$SR = \frac{\Delta T * c}{2}$$

όπου

- ΔT η διάρκεια του παλμού,
- c η ταχύτητα του φωτός στην ατμόσφαιρα



Σχήμα 5: Αντιμετώπιση θορύβου με συσσώρευση φασμάτων

Η ατμόσφαιρα θα χωριστεί τώρα σε περιοχές, ή αλλιώς χωρικά σημεία, για κάθε μία από τις οποίες θα προσδιοριστεί η ταχύτητα του ανέμου. Κάθε χωρικό σημείο θα αποτελείται από έναν αριθμό δειγμάτων, ο οποίος από δω και στο εξής θα αναφέρεται ως μήκος πύλης L_g. Η ανάγκη για ύπαρξη δειγμάτων $L_g > 1$ είναι δεδομένη, καθώς όσο περισσότερα είναι τα δείγματα για κάθε παράθυρο, τόσο περισσότερη πληροφορία υπάρχει και τόσο πιο εύκολα θα γίνει η εξάλειψη του θορύβου για τον εντοπισμό της ταχύτητας του ανέμου στη συγκεκριμένη περιοχή. Εγείρεται εδώ, όμως, μία ανάγκη για συμβιβασμό, καθώς όσο μεγαλύτερο είναι

το μήκος πύλης, τόσο πιο μεγάλη θα είναι και η αβεβαιότητα για το σημείο από το οποίο λαμβάνονται τα δεδομένα.

Όσο δηλαδή μεγαλύτερο είναι το μήκος πύλης, τόσο πιο εύκολα θα εξαλειφθεί ο θόρυβος από το οπισθοσκεδαζόμενο σήμα για τον προσδιορισμό της ταχύτητας του ανέμου, αλλά και τόσο πιο ασαφές θα είναι το μέρος το οποίο αφορά ο υπολογισμός.

Το σύστημα LiDAR αποστέλλει αρχικά παλμό και ύστερα από συγκεκριμένο χρονικό διάστημα στο τηλεσκόπιο του LiDAR καταφθάνουν τα πρώτα οπισθοσκεδασμένα φωτόνια της ακτινοβολίας. Τα L_g πρώτα δείγματα στο τηλεσκόπιο αποτελούν ουσιαστικά το πρώτο παράθυρο δεδομένων προς επεξεργασία. Τα δεδομένα αυτά θεωρείται πως προέρχονται από το πλησιέστερο στο LiDAR "χωρικό σημείο" στην ακτινική διεύθυνση του τηλεσκοπίου, κάτι το οποίο θα αναλυθεί εκτενώς στη συνέχεια. Τα επόμενα L_g δείγματα θεωρείται πως προέρχονται από το 2ο "χωρικό σημείο", τα επόμενα από το 3ο κ.ο.κ μέχρις ότου να φτάσουμε στα δείγματα που προέρχονται από σημείο στα όρια της ακτίνας ανίχνευσης του Wind LiDAR.

Έχοντας αποθηκεύσει τα δεδομένα για κάθε παράθυρο ξεχωριστά, είναι δυνατόν, μεταβαίνοντας στον τομέα της συχνότητας, να ληφθούν τα πρώτα φάσματα για κάθε παράθυρο ξεχωριστά, τα οποία προς το παρόν είναι θορυβώδη. Τώρα, επιτρέπεται στο LiDAR να εκπέμψει το 2ο παλμό, ο οποίος με τη σειρά του θα επιστρέψει για κάθε "χωρικό σημείο" L_g δείγματα. Μετατρέποντας στον τομέα της συχνότητας και αθροίζοντας τα αντίστοιχα φάσματα για κάθε παράθυρο αρχίζει η διαδικασία εξάλειψης του θορύβου.

Κάθε πύλη στην ατμόσφαιρα αποτελείται από L_g δείγματα, έχοντας επομένως πραγματικό μήκος $l = L_g * c * T_{ADC}$. Γνωρίζοντας και την ακτίνα ανίχνευσης του LiDAR, με μία απλή διαίρεση υπολογίζεται ο αριθμός των πυλών της ατμόσφαιρας για τις οποίες μπορεί να προσδιοριστεί η ταχύτητα του ανέμου:

$$N_p = \frac{R_D}{l},$$

όπου

- N_p ο αριθμός των χωρικών σημείων,
- R_D η ακτίνα ανίχνευσης του LiDAR και
- l το πραγματικό μήκος κάθε πύλης της ατμόσφαιρας

Προκύπτουν, λοιπόν, οι εξής απαιτήσεις για ορθή και αποδοτική λειτουργία του συστήματος:

I Επιλογή μήκους πύλης L_g που ικανοποιεί την ζητούμενη ακρίβεια στον υπολογισμό της ταχύτητας, αλλά παραμένει και σχετικά μικρό, ώστε να υπάρχουν χωρικά σημεία μεγάλης ακρίβειας.

II Επιλογή μήκους FFT με σκοπό την ικανοποιητική ακρίβεια υπολογισμού, με το ελάχιστο δυνατό μήκος για εξοικονόμηση υπολογιστικού φόρτου (πόρων, χρόνου, ισχύος).

III Επιλογή πλατφόρμας SoC ώστε να ικανοποιεί τις ανάγκες της σχεδίασης σε μνήμη, αλλά και να υποστηρίζει το ταχύτερο δυνατό ρολόι για το οποίο λειτουργεί η σχεδίαση, με σκοπό την ελαχιστοποίηση του χρόνου εκτέλεσης.

IV Αποδοτική σχεδίαση σε επίπεδο HDL για παράλληλη, pipelined λειτουργία με σωστή αξιοποίηση των πόρων που παρέχει η SoC πλατφόρμα.

Για την ικανοποίηση των απαιτήσεων (I) και (II) πραγματοποιήθηκε μελέτη σε MATLAB, σχετική με την εύρεση των βέλτιστων παραμέτρων. Έχοντας, έπειτα, καταλήξει στις επικρατέστερες, δημιουργήθηκε design σε VHDL με αυτές ως σταθερές και σκοπό την ικανοποίηση και των απαιτήσεων (III) και (IV).

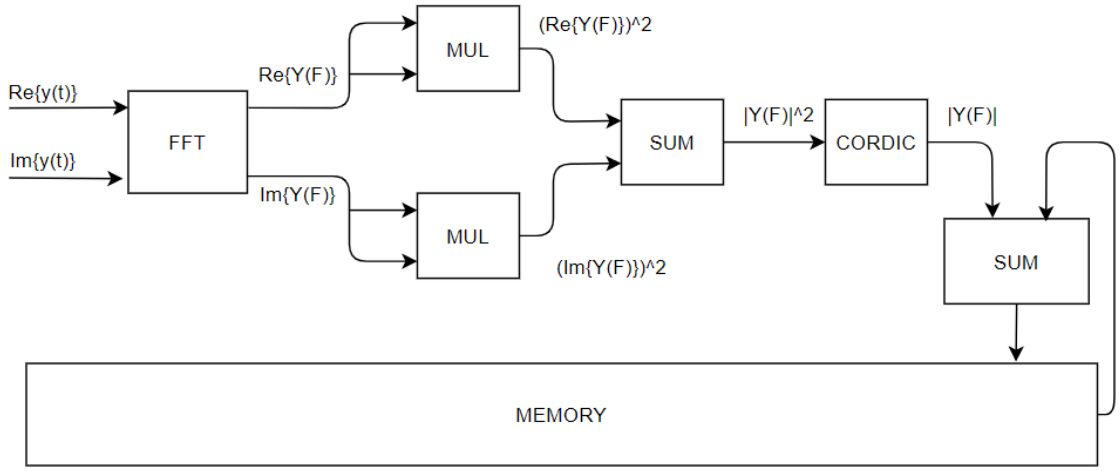
Υλοποίηση Συστήματος Επεξεργασίας

Ανάλυση Συχνοτήτων και Άθροιση

Το PL κομμάτι του συστήματος θα δέχεται στην είσοδό του τα δεδομένα εισόδου του LiDAR, αφού έχουν διακριτοποιηθεί με τη χρήση ADC σε συχνότητα 500MHz. Το σήμα εισόδου αυτό, αποτελείται από μία συνιστώσα μέτρου και μία συνιστώσα γωνίας, είναι δηλαδή μιγαδικό σήμα. Για την ανάλυση συχνότητας των δεδομένων θα χρησιμοποιηθεί ο FFT αλγόριθμος και συγκεκριμένα η υλοποίησή του με το IP της "Xilinx Fast Fourier Transform v9.0 LogiCORE IP". Το συγκεκριμένο component παράγει στην έξοδό του, μεταξύ άλλων, και δύο σήματα που αντιστοιχούν το ένα στο πραγματικό και το άλλο στο φανταστικό μέλος του φάσματος που προκύπτει. Τα δύο αυτά σήματα θα υψωθούν στο τετράγωνο με τη χρήση pipelined πολλαπλασιαστών, θα αθροιστούν και το αποτέλεσμα θα περάσει από το "CORDIC IP" της Xilinx που υπολογίζει την τετραγωνική ρίζα της εισόδου του, δηλαδή το μέτρο κάθε σημείου του φάσματος.

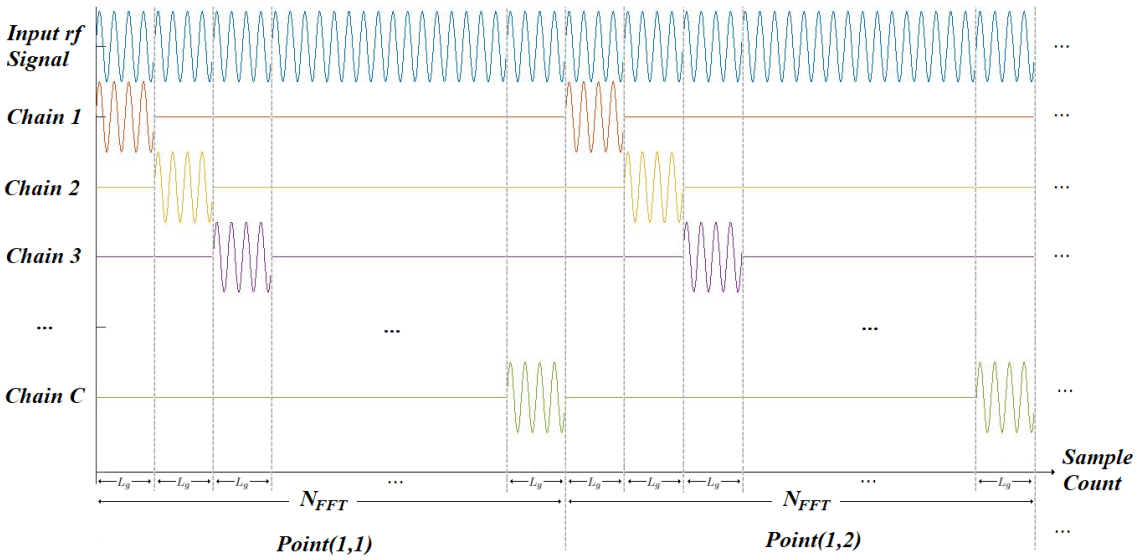
Για την συσσώρευση των φασμάτων των διαφόρων παραθύρων θα χρησιμοποιηθεί Block RAM μνήμη για την αποθήκευση του έως τότε αθροίσματος για το κάθε σημείο και ένας pipelined αθροιστής, ο οποίος θα αθροίζει τα αποθηκευμένα δεδομένα στη μνήμη με τα αντίστοιχα σημεία που προκύπτουν από τον FFT και θα επιστρέφει το αποτέλεσμα με overwrite στη μνήμη.

Δεν αρκεί μόνο η λειτουργικότητα της σχεδίασης, απαιτείται επίσης και η "έξυπνη" και αποδοτική λειτουργία της. Στόχος είναι σε κάθε κύκλο ρολογιού να εκτελούνται υπολογιστικές διεργασίες και ποτέ να μην υπάρξει component που δε χρησιμοποιείται, εφόσον υπάρχουν βέβαια δεδομένα προς επεξεργασία. Για να συμβεί αυτό, όλα τα components που χρησιμοποιούνται στο design έχουν επιλεχθεί ή έχουν σχεδιαστεί να λειτουργούν πλήρως pipelined. Επειδή σε ένα FFT-component δίνονται L_g δείγματα εισόδου και έπειτα μηδενικά μέχρι να συμπληρωθούν N_{FFT}, όσα δηλαδή και το μήκος του, αν χρησιμοποιηθεί στο σύστημα μία μόνο "άλυσίδα" επεξεργασίας, τότε θα πρέπει να καθυστερήσουν ή ακόμα και να αγνοηθούν τα επόμενα $N_{FFT}-L_g$ δείγματα εισόδου, κάτι το οποίο δεν είναι αποδοτικό. Για να αξιοποιηθεί στο μέγιστο η συχνότητα δειγματοληψίας του συστήματος, θα πρέπει να επιλεγεί μήκος πύλης L_g που να διαιρεί ακριβώς το μήκος N_{FFT} και να χρησιμοποιηθούν $C = N_{FFT}/L_g$ "άλυσίδες"



Σχήμα 6: Αλυσίδα επεξεργασίας δεδομένων

επεξεργασίας στο design. Με τον τρόπο αυτό, ένα arbiter κύκλωμα θα αναθέτει κάθε δείγμα εισόδου A_i στην $((A_i \div L_g) \bmod N_{FFT})$ "άλυσίδα" επεξεργασίας, ενώ θα δίνει μηδενικά ως εισόδους στις υπόλοιπες $(C-1)$ αλυσίδες. Αποδεικνύεται θεωρητικά και μελετήθηκε και πρακτικά με πειράματα σε MATLAB ότι τα αποτελέσματα στην ανίχνευση της επικρατούσας συχνότητας, μέσω συχνοτικής ανάλυσης με το μετασχηματισμό Fourier, είναι τα ίδια, ανεξάρτητα από την περιοχή του σήματος στην οποία γίνονται pad τα μηδενικά. Εκμεταλλευόμενοι το γεγονός, είναι δυνατόν σε N_{FFT} κύκλους ρολογιού να τροφοδοτηθούν όχι ένας, αλλά C FFTs με τα σωστά σήματα εισόδου και το κύκλωμά να λειτουργεί πλήρως pipelined και να έχει throughput C αντί για 1.



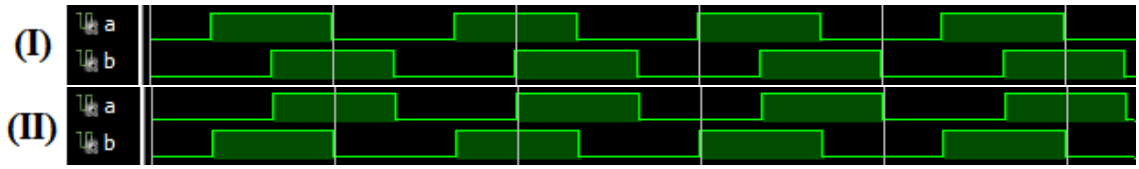
Σχήμα 7: Κατανομή δειγμάτων εισόδου στους επεξεργαστικούς πόρους

Το LiDAR εκπέμπει αρχικά ημιτονοειδή παλμό για κάποιο χρονικό παράθυρο και, έπειτα από κάποιο χρόνο, ανιχνεύεται το πρώτο δείγμα της οπισθοσκεδαζόμενης ακτινοβολίας, όπως φαίνεται στο Σχήμα 7. Αυτό καθώς και τα επόμενα $L_g - 1$ θα δοθούν, αφού ψηφιοποιηθούν

μέσω του ADC, ως είσοδοι στην πρώτη αλυσίδα επεξεργασίας και αντιστοιχούν στο πρώτο χωρικό σημείο που μελετάται. Τα επόμενα L_g δείγματα, που αντιστοιχούν στο 2ο χωρικό σημείο, θα δοθούν στη δεύτερη αλυσίδα κ.ο.κ. μέχρις ότου να συμπληρωθούν τα N_{FFT} πρώτα δείγματα. Στον επόμενο κύκλο ρολογιού, σε όλους τους FFT θα έχουν δοθεί N_{FFT} δείγματα, αρκετά ώστε να δημιουργηθεί το φάσμα συχνοτήτων εξόδου και, αφού λειτουργούν pipelined, θα είναι όλοι έτοιμοι να δεχθούν δεδομένα για το επόμενο χωρικό σημείο. Προφανώς, τα επόμενα L_g δείγματα θα δοθούν πάλι στην 1η αλυσίδα επεξεργασίας, θα αντιστοιχούν τώρα όμως στο χωρικό σημείο $C + 1$. Εγείρεται λοιπόν η ανάγκη, κάθε αλυσίδα επεξεργασίας να διαθέτει μνήμη που θα επιτρέπει το άθροισμα όχι μόνο για ένα, αλλά για περισσότερα χωρικά σημεία. Έχοντας, μάλιστα, υπολογίσει τον αριθμό χωρικών σημείων N_P, θα πρέπει αυτά να μοιραστούν στις αλυσίδες επεξεργασίας. Συγκεκριμένα, για C αλυσίδες κάθε αλυσίδα θα έχει χώρο μνήμης για αποθήκευση N_P/C (ή $N_P/C + 1$ αν η διαίρεση έχει υπόλοιπο) φασμάτων. Κάθε χωρικό σημείο G_i θα αντιστοιχεί στο $G_i \div C$ φάσμα της $G_i \bmod C$ αλυσίδας επεξεργασίας.

Διεύθυνση Ανίχνευσης

Το Coherent Detection Doppler LiDAR του συστήματος στρέφεται από δύο κινητήρες που μεταβάλλουν το αζιμούθιο και τη γωνία ανύψωσής του. Κατά την περιστροφή, οι κωδικοποιητές των κινητήρων παράγουν δύο σήματα του 1 bit, ο συνδυασμός των οποίων μπορεί να πληροφορήσει τόσο για την φορά όσο και για τις μοίρες της περιστροφής. Η συχνότητα μεταβολής αυτών των ψηφιακών σημάτων εξαρτάται από τη γωνιακή ταχύτητα περιστροφής και δεν ξεπερνάει τα 15 kHz. Οι μεταβολές των σημάτων, έστω a και b, για περιστροφή (I) κατά τη φορά του ρολογιού και (II) αντίθετα φαίνονται στο Σχήμα 8.



Σχήμα 8: Σήματα του κινητήρα για την περιστροφή στην (I) ωρολογιακή και (II) αντιωρολογιακή φορά

Οι μεταβολές καταμετρώνται με ένα κύκλωμα μετρητή στο FPGA σύμφωνα με τον Πίνακα Αληθείας 1. Τα σύμβολα ¨+' και ¨-' αντιστοιχούν σε αύξηση και μείωση αντίστοιχα του μετρητή κατά 1, ενώ το ¨/' σημαίνει καμία μεταβολή. Η τιμή του μετρητή μετατρέπεται στον ARM σε μοίρες με πολλαπλασιασμό με σταθερά που εξαρτάται από τις αναλογίες των γραναζιών των κινητήρων. Το κύκλωμα αυτό στο PL μπορεί να λειτουργήσει και με ρολόι με χαμηλότερη συχνότητα από το υπόλοιπο κύκλωμα για περαιτέρω μείωση της κατανάλωσης.

old A	A	old B	B	cnt
0	0	0	0	/
0	0	0	1	-
0	0	1	0	+
0	0	1	1	/
0	1	0	0	+
0	1	0	1	+
0	1	1	0	-
0	1	1	1	-
1	0	0	0	-
1	0	0	1	-
1	0	1	0	+
1	0	1	1	+
1	1	0	0	/
1	1	0	1	+
1	1	1	0	-
1	1	1	1	/

Πίνακας 1: Πίνακας αληθείας για κύκλωμα μετρητή γωνίας

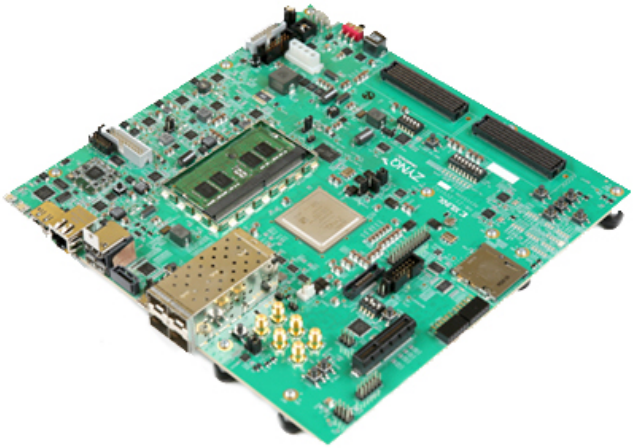
Zynq Ultrascale 102+ MPSoC

Η ανάγκη του συστήματος για υπολογιστικούς πόρους και θέσεις μνήμης, σε συνδυασμό με την επιθυμία για αξιόπιστη, άμεση επικοινωνία με το περιβάλλον και για ευελιξία για μελλοντικές μεταβολές ή προσθήκη περισσότερων κυκλωμάτων, οδηγούν στην επιλογή ενός ισχυρού, σύγχρονου multi-processor SoC.

Το Zynq Ultrascale 102+ (ZCU102) της Xilinx περιέχει έναν τετραπύρηνο Arm® Cortex®-A53, έναν διπύρηνο Cortex-R5F real-time επεξεργαστή και μία μονάδα γραφικών Mali™-400 MP2. Υποστηρίζει τα πιο κύρια και διαδεδομένα περιφερειακά και διασυνδέσεις, επιτρέποντας τη σχεδίαση για μεγάλη ποικιλία εφαρμογών.

Επιτρέπει διαμόρφωση μέσω USB (onboard JTAG), dual Quad-SPI flash memory ή SD κάρτα. Το PS διαθέτει 4GB DDR4 64-bit SODIMM, ενώ το PL μνήμη 512MB DDR4 ([256 Mb x 16] devices) στα 1200MHz / 2400Mbps DDR. Διαθέτει μεταξύ άλλων και IIC EEPROM των 8KB και RJ45 Ethernet connector.

Κάποια χαρακτηριστικά του MPSoc Zynq UltraScale+ XCZU9EG-2FFVB1156 δίνονται στον Πίνακα 2.



Σχήμα 9: Zynq Ultrascale 102+ MPSoC [33]

System Logic Cells (K)	600
Memory (Mb)	32.1
DSP Slices	2520
Maximum I/0 Pins	328

Πίνακας 2: Χαρακτηριστικά ZCU102

5. Πειραματικά αποτελέσματα

Μοντελοποίηση Διαδικασίας Ανίχνευσης σε MATLAB

Σε MATLAB πραγματοποιήθηκαν προσομοιώσεις της διαδικασίας υπολογισμού της ταχύτητας του ανέμου από δεδομένα Wind LiDAR (CDWL), με σκοπό τη διερεύνηση της επίδρασης των παραμέτρων που την επηρεάζουν στην ακρίβεια του υπολογισμού. Οι προσομοιώσεις βασίζονται σε παραδοτέο κώδικα που αναπτύχθηκε για το project ¨ΑΙΟΛΙΚΟΣ' από το Microlab ECE NTUA . Οι παράμετροι είναι το μήκος πύλης L_g, το μήκος (αριθμός σημείων) του FFT N_{FFT}, καθώς και ο αριθμός accumulations, έστω P, που χρειάζεται για να υπάρχει ικανοποιητική ακρίβεια υπολογισμού για τους διάφορους συνδυασμούς των προηγουμένων δύο.

Το φάσμα συχνοτήτων στο σύστημα μπορεί να παρασταθεί από δύο συνιστώσες, μία Rayleigh και μία Mie, με την προσθήκη Λευκού Γκαουσιανού Θορύβου. Οι συνιστώσες αυτές δημιουργήθηκαν έτσι ώστε να ρυθμίζονται σε διάφορα επίπεδα, για την προσομοίωση διαφορετικών καταστάσεων θορύβου. Άλλες πιθανές αιτίες ανακριβειών προσομοιώνονται εισάγοντας διακύμανση στην ένταση I_{var} της οπισθοσκεδαζόμενης ακτινοβολίας, αλλά και στη συχνότητα f_{var} των λαμβανόμενων παλμών, συνθήκες πιθανές σε ένα πραγματικό σύστημα LiDAR.

Για την προσομοίωση επιλέγεται μία συχνότητα F_{pul} για την εκπεμπόμενη ακτινοβολία. Σε MATLAB μοντελοποιείται η θορυβώδης οπισθοσκεδαζόμενη ακτινοβολία που ανιχνεύεται. Δημιουργείται, αρχικά, ένα ημιτονοειδές σήμα μήκους L_g δειγμάτων, με συχνότητα τυχαία παραλλαγμένη γύρω από την F_{pul} και με τυχαία ένταση, ανάλογα με τα f_{var}, I_{var}. Το σήμα τοποθετείται τυχαία σε κάποια θέση ενός μηδενικού διανύσματος μήκους N_{FFT} (κάνοντας ουσιαστικά zero padding πριν και μετά), με σκοπό να δοθεί ως είσοδος σε έναν FFT μήκους N_{FFT}. Πριν γίνει αυτό, προστίθεται ένα επίπεδο Λευκού Γκαουσιανού Θορύβου με τη συνάρτηση AWGN() του MATLAB. Το σήμα σε αυτό το σημείο είναι έτοιμο να δοθεί στον FFT για φασματική ανάλυση. Η ανάλυση μίας μόνο θορυβώδους εισόδου L_g σημείων οδηγεί σε ανακριβή εντοπισμό της επικρατούσας συχνότητας (Σχήμα 4). Η διαδικασία αυτή επαναλαμβάνεται P φορές, δημιουργώντας κάθε φορά ένα καινούργιο, τυχαία διαφοροποιημένο από το τέλειο ημίτονο σήμα εισόδου και τα φάσματα εξόδου αθροίζονται.

Από τα φάσματα που παράγονται προκύπτει η συχνότητα ανίχνευσης. Για να συμβεί αυτό, εντοπίζεται η μέγιστη Πυκνότητα Ισχύος του Φάσματος που έχει δημιουργηθεί. Δεν αρκεί, βέβαια, η υπόθεση πως η αντίστοιχη συχνότητα είναι και η βασική συχνότητα της οπισθοσκεδαζόμενης ακτινοβολίας. Αυτό συμβαίνει διότι το σύστημα θα μπορούσε έτσι να ανιχνεύσει μόνο N_{FFT} διακριτές συχνότητες, αδυνατώντας να εντοπίσει με ακρίβεια άπειρες πιθανές συχνότητες του συνεχούς πεδίου ορισμού της F_{pul}. Για προσέγγιση με μεγαλύτερη ακρίβεια ακολουθείται η τεχνική της προσαρμοσμένης καμπύλης (curve fitting). Πρόκειται για την τεχνική προσέγγισης μίας σειράς δεδομένων με μία συνάρτηση ή καμπύλη. Συνήθης είναι η χρήση πολυωνυμικών συναρτήσεων. Στη συγκεκριμένη περίπτωση για την εφαρμογή του curve fitting δίνονται ως δεδομένα το ζεύγος της μέγιστης τιμής του φάσματος με τη συχνότητα στην οποία αντιστοιχεί, καθώς και τα ζεύγη τιμής-συχνότητας που αντιστοιχούν στην ακριβώς προηγούμενη και την ακριβώς επόμενη συχνότητα του φάσματος. Επιλέχθηκε, χάριν απλούστευσης, η χρήση παραβολοειδούς καμπύλης, ή αλλιώς πολυωνύμου $2^{ου}$ βαθμού.

Αν A, B, C τα διαδοχικά σημεία του φάσματος συχνοτήτων που προαναφέρθηκαν, όπου B το σημείο του μεγίστου και A και C το προηγούμενο και επόμενο δείγμα αντίστοιχα, προκύπτει, παραγωγίζοντας το πολυώνυμο δευτέρου βαθμού και βρίσκοντας το σημείο μηδενισμού της παραγώγου [34], μέγιστο στη συχνότητα $f_d = f_B + \Delta f$, όπου

$$\Delta f = \frac{f_A - f_C}{2(f_A + f_C - 2f_B)}$$

επομένως

$$f_d = f_B + \frac{f_A - f_C}{2(f_A + f_C - 2f_B)}$$

Με τον τρόπο αυτό, το διακριτό σύνολο αφίξεως του συστήματός διευρύνεται. Όταν πλέον το P είναι επαρκώς μεγάλο, είναι δυνατόν να υπάρξουν επίπεδα μεγάλης ακρίβειας στην ταύτιση της συχνότητας ανίχνευσης f_d με τη συχνότητα εισόδου F_{pul}.

Με βάση την λειτουργία του κώδικα που επεξηγήθηκε, εκτελέστηκαν προσομοιώσεις για διάφορες τιμές των παραμέτρων προς βελτιστοποίηση και των μεταβλητών που τις επηρεάζουν. Απαραίτητη ορίστηκε η επίτευξη ακριβείας μεγαλύτερης του 0.1 MHz ως προς τον εντοπισμό

της συχνότητας F_{pul}, κάτι που θα οδηγήσει τελικά σε ακρίβεια μεγαλύτερη του 0.1 m/s στην ανίχνευση της ταχύτητας του ανέμου.

Καταρχάς επιλέχθηκε για τις προσομοιώσεις αλλά και το τελικό σύστημα μήκος $N_{FFT} = 1024$ δείγματα, μία δύναμη του 2 για να αξιοποιηθεί πλήρως η δομή του Fast Fourier Transform. Παρατηρήθηκε σε κάποια πρώτα πειράματα σε MATlAB, πως για μικρότερες τιμές του N_{FFT} (πχ 256, 512), ακόμα και για μηδενικό θόρυβο ($SNR = 100dB$) η συχνότητα f_d που ανιχνευόταν δεν προσέγγίζε ικανοποιητικά την επιθυμητή συχνότητα F_{pul}. Για $N_{FFT} \geqslant 1024$ προέκυψε προσέγγιση σε ικανοποιητικά επίπεδα σε αρχικά στάδια, οπότε επιλέχθηκε η τιμή 1024, για εξοικονόμηση χρόνου και πόρων που θα απαιτούσαν οι μεγαλύτερες τιμές N_{FFT}, εις βάρος όμως της μεγαλύτερης ευκρίνειας που θα επέτρεπαν αυτές.

Οι προσομοιώσεις που ακολούθησαν διερεύνησαν δύο κύρια ζητήματα:

I τον προσδιορισμό της σχέσης του μήκους πύλης με την ακρίβεια μέτρησης και

II τον προσδιορισμό του ελάχιστου αριθμού accumulations P που απαιτείται για κάθε συνδυασμό παραμέτρων για να επιτευχθεί η ζητούμενη ακρίβεια υπολογισμού, για διάφορες συνθήκες θορύβου.

Σημειωτέον ότι για τον υπολογισμό του απόλυτου σφάλματος μέτρησης επαναλαμβάνεται η προαναφερθείσα διαδικασία rep φορές και υπολογίζεται η απόσταση της μέσης υπολογισμένης συχνότητας $\overline{f_d}$ από τη συχνότητα f_{pul}.

$$\overline{f_{error}} = |F_{pul} - \overline{f_d}|$$

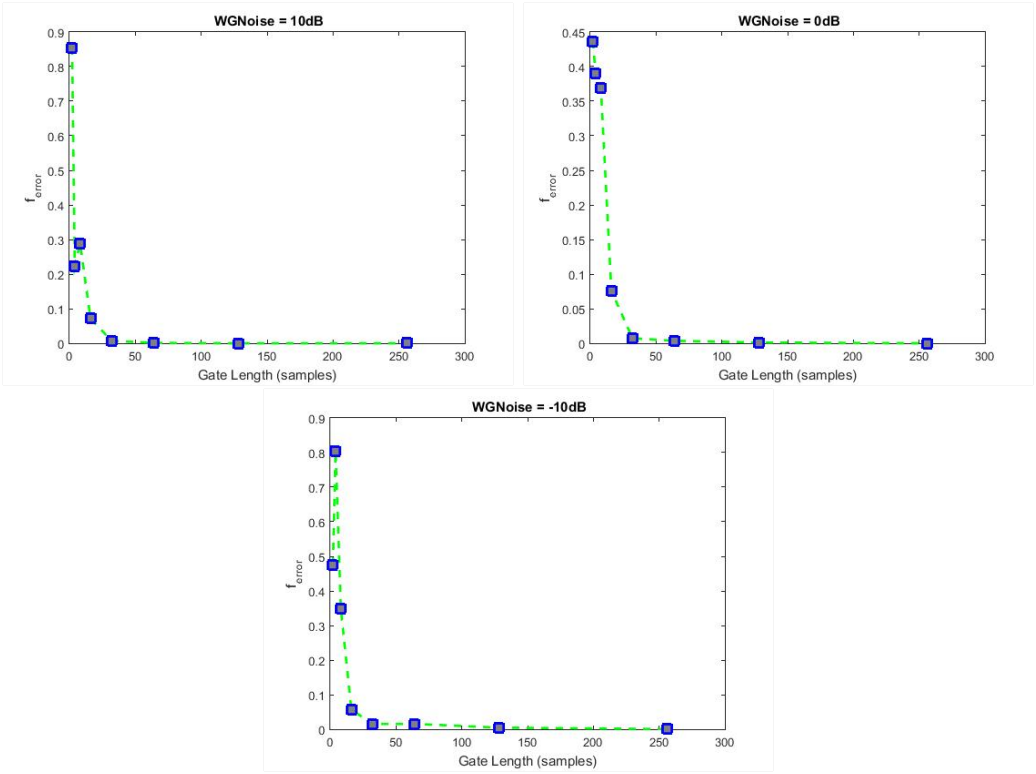
όπου

$$\overline{f_d} = \frac{\sum_{i=1}^{rep}(f_d)_i}{rep}$$

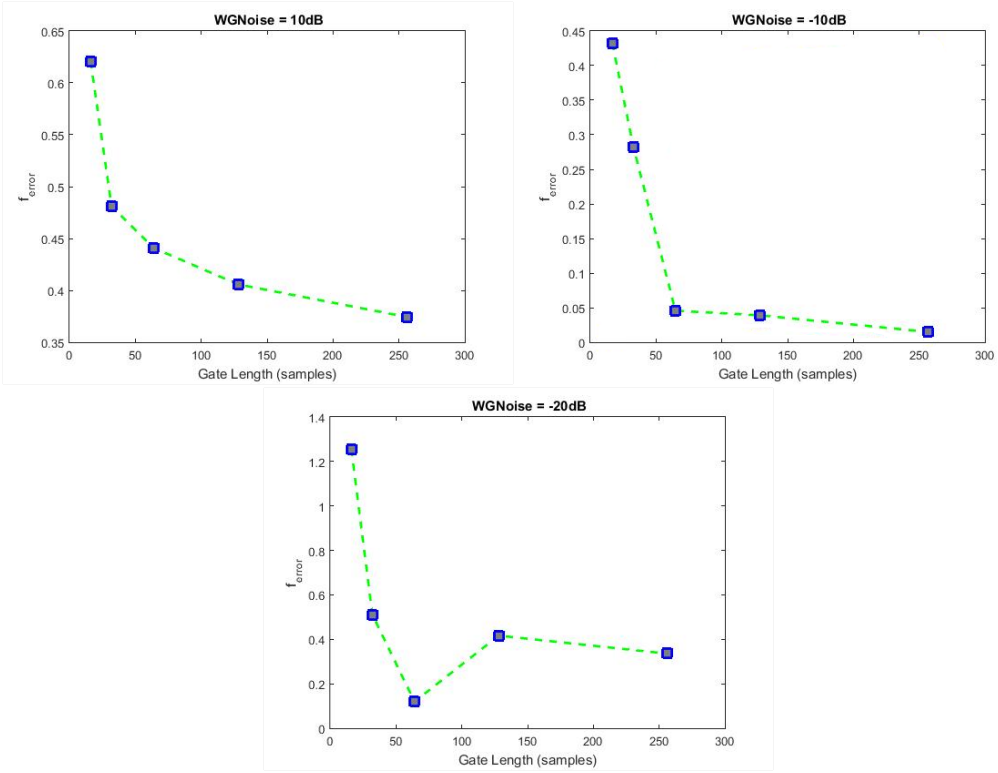
Επίδραση του μήκους πύλης στην ακρίβεια της μέτρησης

Στα πλαίσια αυτής της προσομοίωσης υπολογίστηκε για διάφορα επίπεδα θορύβου η επίδραση διαφορετικών μηκών πύλης στο απόλυτο σφάλμα μέτρησης, για σταθερό αριθμό accumulations P. Τα Σχήματα 10 και 11 περιέχουν αποτελέσματα προσομοιώσεων για 512 και για 2048 accumulations, για $rep = 1000$ επαναλήψεις.

Τα γραφήματα επιβεβαιώνουν την εξάρτηση της ποιότητας των αποτελεσμάτων από το μήκος πύλης της διάταξης, ανεξαρτήτως θορύβου. Οποιαδήποτε spikes παρατηρούνται, οφείλονται στο μικρό αριθμό επανάληψης του πειράματος rep και αφορούν μόνο μικρές διαφορές του σφάλματος μέτρησης. Επιλέγονται τελικά για τη σχεδίαση του συστήματος τα μήκη πύλης $L_g = 32$ και $L_g = 64$, αφού σε όλα τα γραφήματα μεταβάλλεται απότομα η κλίση που αντιστοιχεί σε μείωση του σφάλματος μέτρησης είτε μετά το $L_g = 32$ είτε μετά το $L_g = 64$.



Σχήμα 10: $f_{error} = f(L_g)$, για P = 512 accumulations

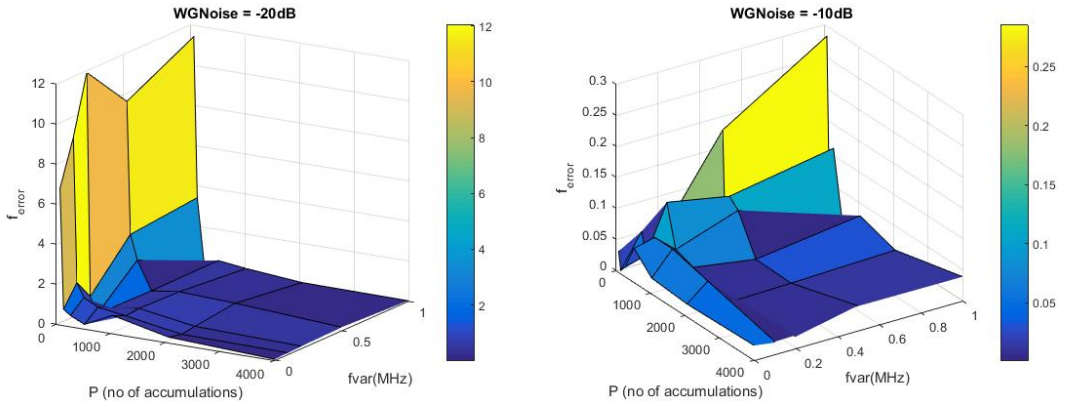


Σχήμα 11: $f_{error} = f(L_g)$, για P = 2048 accumulations

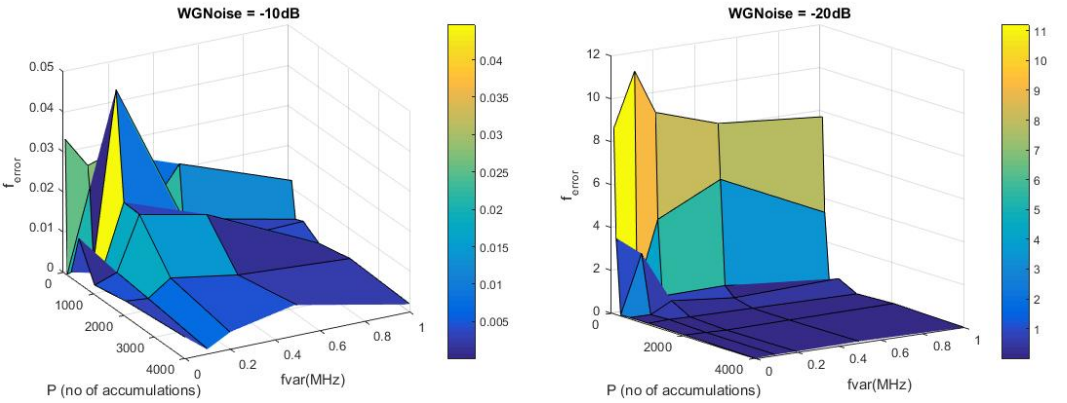
Υπολογιστικό σφάλμα συναρτήσει του αριθμού accumulations για διάφορες συνθήκες θορύβου

Με βάση τα αποτελέσματα των πρώτων προσομοιώσεων, η μελέτη συνεχίζεται με χρήση μήκους πύλης L_g 32 και 64 δειγμάτων. Για κάθε μία από τις δύο περιπτώσεις είναι ζητούμενο να εκτιμηθεί ο αριθμός των accumulations P που οδηγεί σε αποδεκτό σφάλμα μέτρησης, ανάλογα με το επίπεδο του θορύβου. Για το σκοπό αυτό εκτιμήθηκε για $rep = 1000$ επαναλήψεις της διαδικασίας, το σφάλμα μέτρησης για μεγάλο εύρος P για τις δύο περιπτώσεις μήκους πύλης L_g. Σε πρώτο στάδιο η διαδικασία εκτελέστηκε για διακριτά επίπεδα Λευκού Γκαουσιανού Θορύβου -10dB και -20dB, για ζεύγη $[f_{var}, I_{var}] = [0.2$ MHz, $20\%]$ και $[1$ MHz , $50\%]$, ενώ στη συνέχεια ακολούθησε προσομοίωση για περισσότερες τιμές της μεταβλητής f_{var} και τρισδιάστατη απεικόνιση των αποτελεσμάτων.

Τα αποτελέσματα των προσομοιώσεων του πρώτου σταδίου περιέχονται στον Πίνακα 3 και η 3-D απεικόνιση στα Σχήματα 12 και 13.



Σχήμα 12: $f_{error} = f(P, f_{var})$ με $L_g = 32$



Σχήμα 13: $f_{error} = f(P, f_{var})$ με $L_g = 64$

AWGN	**-10 dB**		**-20 dB**	
(I_{var},f_{var})	(0.2,0.2)	(0.5,1)	(0.2,0.2)	(0.5,1)
$L_g = 32$	2630 accs	3010 accs	9550 accs	15260 accs
$L_g = 64$	330 accs	620 accs	3900 accs	6020 accs

Πίνακας 3: Απαιτούμενος αριθμός μετρήσεων για την επίτευξη ακρίβειας 0.1 MHz

VHDL Design

Βάσει των αποτελεσμάτων των προσομοιώσεων σε MATLAB και της ανάλυσης του Κεφαλαίου 4 αναπτύχθηκαν σε VHDL δύο συστήματα για την υλοποίηση της ανάλυσης συχνότητας και άθροισης φασμάτων, ένα με $L_g = 32$ και ένα με $L_g = 64$ δείγματα. Τα μεγέθη που επιλέχθηκαν για το design προέκυψαν από τους τύπους του Κεφαλαίου 4 και οι πόροι που απαιτούνται για την υλοποίησή τους, μεταξύ των πόρων του Zynq Ultrascale 102+ (274080 LUTs, 144000 LUTRAMs, 548160 FFs, 912 BRAMs, 2520 DSPs), καθώς και η χρονικές επιδόσεις των δύο κυκλωμάτων βρίσκονται στους Πίνακες 4 και 5:

	CONSTANTS				**RESOURCES**				
L_g	N_{FFT}	C	N_P	$\frac{Spectra}{Chain}$	LUT	LUTRAM	DFF	DSP	BLOCKRAM
32	1024	32	512	16	32%	19%	32%	51%	63%
64	1024	16	256	16	16%	10%	16%	25%	32%

Πίνακας 4: Χρήση Πόρων

	CONSTANTS				**TIMING PERFORMANCE**		
L_g	N_{FFT}	C	N_P	$\frac{Spectra}{Chain}$	Clock Frequency (MHz)	Processing Time per accumulation (μs)	Execution Time Max Accumulations Zero Wait (ms)
32	1024	32	512	16	500	35.08	535.32
64	1024	16	256	16	500	35.08	211.18

Πίνακας 5: Χρονική Επίδοση

Έχει θεωρηθεί ακτίνα ανίχνευσης περίπου 10 km και $T_{ADC} = 2ns$. Ο χρόνος εκτέλεσης έχει υπολογιστεί για κάθε design για το μέγιστο αριθμό accumulations που προέκυψε από τις προσομοιώσεις MATLAB (Πίνακας 3).

6. Συμπεράσματα και μελλοντική έρευνα

Συμπερασματικά, η διπλωματική αυτή εργασία επιχείρησε την ανάπτυξη συστήματος για την ψηφιακή επεξεργασία και τον έλεγχο σε Coherent Detection Doppler LiDAR και τη βελτιστοποίηση της διαδικασίας υπολογισμού της μέσης τιμής των μετρήσεων για προσδιορισμό της ταχύτητας του ανέμου. Εκτιμήθηκε για αυτό το σκοπό ο απαιτούμενος αριθμός

μετρήσεων προς άθροιση και επομένως ο απαιτούμενος χρόνος εκτέλεσης υπολογισμών για τον προσδιορισμό της ταχύτητας του ανέμου σε ορισμένα σημεία της ατμόσφαιρας στην ακτινική διεύθυνση του LiDAR. Αναπτύχθηκαν δύο προτεινόμενα συστήματα επεξεργασίας σε SoC FPGA με διαφορετικό χρόνο εκτέλεσης και κατανάλωση πόρων, προσφέροντας έτσι και διαφορετική διαβάθμιση στην χωρική ακρίβεια των μετρήσεων. Σε περιπτώσεις που ο χρόνος είναι σημαντικός, το γρηγορότερο από τα δύο design μπορεί να αποτελέσει βάση για το υπολογιστικό σύστημα σε FPGA, ενώ σε εφαρμογές που απαιτούν μεγαλύτερη ακρίβεια στη χωρική ανάλυση μπορεί να χρησιμοποιηθεί η δεύτερη πρόταση με το αντίστοιχο κόστος σε χρόνο εκτέλεσης. Η χρήση μεταβλητών μεγεθών για την υλοποίηση του συστήματος επιτρέπει και τη μείωση του αριθμού μετρήσεων που επεξεργάζεται το σύστημα για να προσδιορίσει την ταχύτητα του ανέμου, όταν το επιτρέπουν οι συνθήκες θορύβου, με σκοπό τη μείωση του χρόνου εκτέλεσης.

Βεβαίως, υπάρχει χώρος για βελτίωση και περαιτέρω έρευνα. Η προτεινόμενη σχεδίαση έχει ελεγχθεί πειραματικά μόνο με δεδομένα προσομοίωσης, χωρίς tuning μεταξύ ADC και SoC. Πρώτα αυτό, έπειτα η χρήση πραγματικών μετρήσεων LiDAR και η ενσωμάτωση σε πραγματικό σύστημα για μετρήσεις στο πεδίο είναι βήματα που θα πρέπει να γίνουν. Μένει ακόμα η ανάπτυξη και η δοκιμή των υπόλοιπων δύο μελών του προτεινόμενου συστήματος. Η IIC επικοινωνία πρέπει να ελεγχθεί με τους κινητήρες περιστροφής του LiDAR και μια GUI εφαρμογή να αναπτυχθεί σε software επίπεδο για την απομακρυσμένη επικοινωνία με το ψηφιακό σύστημα. Σχετική μελλοντική έρευνα αφορά την υλοποίηση και άλλων αλγορίθμων για ενσωμάτωση στο σύστημα. Υπάρχουν σύγχρονες προτάσεις για φασματική μελέτη των δειγμάτων από Wind LiDAR χωρίς τη χρήση της ανάλυσης Fourier. Η υλοποίηση διαφορετικών αλγορίθμων σε επίπεδο hardware συνιστά, ασφαλώς, αντικείμενο μεγάλου ενδιαφέροντας, προσφέροντας πιθανώς πλεονεκτήματα όπως περαιτέρω επιτάχυνση της διαδικασίας και βελτίωση της ακρίβειας μέτρησης. Τέλος, στα πλαίσια του παρόντος έχει θεωρηθεί ίδια συχνότητα για παροχή δεδομένων από τον ADC και επεξεργασία από το FPGA. Η ανάπτυξη συστήματος που μπορεί να ανταπεξέλθει σε διαφορετικές συχνότητες, καθώς και η εύρεση της μέγιστης δυνατής συχνότητας λειτουργίας αυτού, είναι σημαντικά ζητήματα προς διερεύνηση.

Chapter 1

Introduction

1.1 LiDAR and Applications

The LiDAR (LIght Detection And Ranging) method is a type of active, optic remote sensing. Active sensing is based on the following principle: a source generates radiation to be emitted to the atmosphere and a receiver collects the backscattered (by various aerosols and molecules of the atmosphere) radiation to be analyzed. Remote sensing advantages include independence of time of day and natural radiation sources in general, high intensity and control of stimulating signal and reduced sensitivity to background light.

LiDAR sensing is optic, meaning than in contrast to other active sensing techniques - such as SONAR (SOund NAvigation Ranging) , SODAR (SOund Detection And Ranging), RADAR (RAdiowave Detection And Ranging) - it concerns radiation in the optical frequency range (photons). First appearing in the 1930s and eventually evolved in the early 1960s, right after the invention of laser, LiDAR was initially created for aerosol monitoring and later used for atmospheric particles discrimination, as well as temperature and wind measurements.

The wavelength range available for use, combined with the possibility of accurate laser frequency selection, led to the development of various LiDAR systems. Such are the [15]

- Resonance Fluorescence LiDAR, used to measure temperature and wind density and to track clouds in mid-upper atmosphere
- Differential Absorption LiDAR or DIAL, used to detect ozone, humidity and gaseous pollutants
- Raman LiDAR, also used to measure humidity and to determine the optical density and temperature of the lower atmosphere
- Coherent Doppler LiDAR, used to research wind and turbulence
- Direction Detection LiDAR, for direction tracking applications, and many more.

The above are nowadays commonly used in applications such as geodesy, geomatics, archaeology, geography, geology, geomorphology, seismology, forestry, atmospheric physics, laser guidance, laser altimetry, even for control and navigation of autonomous cars.

This thesis is centered around the Coherent Detection Doppler LiDAR. This system, in short, detects the frequency shift of emitted radiation, backscattered by particles of the atmosphere because of their motion and, exploiting the Doppler Effect formula, determines the particles' and hence the wind's velocity. The accuracy and range provided by such systems has led to their use for wind velocimetry in a variety of applications.

Scientific foundations such as NASA with its conical-scanning HARLIE LiDAR and ESA with the intention of equipping its wind mission "ADM-Aeolus" with a Doppler LiDAR system use this technology for measurements of vertical wind profiles. Apart from its contribution to research in the fields of physics and astronomy, another Wind LiDAR's domain of application is wind farm optimization. LiDARs can improve wind farm energy output, precisely measuring wind speeds. Mounted on wind turbines or integrated into the rotating spinners they can also be used for protection, proactively adjusting blades to the components, when their measurements indicate this is needed. Wind LiDARs are not only used for the optimization of existing wind farms, but they can also be used in deciding the optimal placement of new wind farms. Their measurements along with computational fluid dynamics can lead to the assessment of wind potential.

Of course, Coherent Detection Wind LiDAR data can be applied to other fields, too. Meteorologic predictions concerning wind profiling may be based on the existent technology. Wind measurements are also important in both civil aviation and air force. Wind profiling using a Doppler LiDAR system on a van was even used for wind measurements in the Olympic Games of 2008 in Beijing. Sailors of the Olympic yacht competition benefited from sea-surface wind information produced by a LiDAR system, adjusting their strategies accordingly.

1.2 Problem Statement

The challenge of processing coherent detected LiDAR data is twofold, concerning both heavy Digital Signal Processing and storage of a large amount of data. Although Doppler Wind LiDAR measurements do eventually reach very high levels of accuracy, a single measurement is not enough. Both external and internal sources, as well as the nature of the signal received itself lead to extremely low SNR (Signal to Noise Ratio). This renders the results emerging from the processing of a single pulse received unreliable.

The most common method to solve this problem is averaging. The noise of the measurements can be dealt with by combining the results of multiple scans to exploit noise distribution properties. This, however, requires the storage and processing of a large amount of measurement data, increasing the system's needs for memory resources and the computation time required.

A problem now emerging form the averaging solution, from the scope of hardware, is

data storage. Considering the rate at which the LiDAR system's ADC can provide data, given the development of commercial giga-sample per second converters for integer size samples, the usage of the appropriate hard drive is crucial. Old generation storage means may even be unable to handle such rates of data flow.

Even the usage of state-of-the-art SSDs, though, is probably not an optimal solution. The reasons for that are the increase of the system's control complexity as well as the design's financial cost. Another relevant issue is data reduction. Storing heavy loads of noisy, unreliable data, with only a small percentage of them leading to meaningful information, does not exploit the usage of this costly and advanced technology to the fullest.

All these factors must be taken into consideration when designing the digital system to process Coherent Detection Wind LiDAR data and optimizations may only be achieved by research and experimenting.

1.3 Thesis Scope

The scope of this diploma thesis is the proposal of a system which will execute the digital processing of Doppler Wind LiDAR data for wind velocity computation and will control the LiDAR system's detection direction. The system will be based on experimental results for the optimization of its parameters, in the pursuit of both computation time and resources utilization reduction.

The main research carried out concerns the averaging method. A first stage includes the exploration of noise and the relationship between the system's spatial and temporal resolution to its detection accuracy. A second consideration concerns the trade-off between this accuracy and the received samples' reduction. This requires the determination of the minimum number of measurements in order for the accuracy achieved to be acceptable.

In the direction of resources optimization, a SoC FPGA is going to be used for the implementation. Except for the application-specific, parallel and pipelined operation it can offer, the usage of a SoC can lead to the treatment of more of the issues presented in Paragraph 1.2. The goal is the development of an efficient, fast and accurate real-time processing system on an FPGA platform to totally avoid the need for extra data storage and post-processing. This can lead to rid the processor of time-crucial heavy computations, improve execution time and reduce cost and power consumption. The usage of a modern SoC platform also renders the digital system compact, easy to transport and embed to any Wind LiDAR system. The ARM Processor and the SoC platform peripherals can be used for the communication with the LiDAR system and the development of remote user-friendly interaction.

Chapter 2

Wind LiDAR

2.1 Wind LiDAR Fundamentals

Of all the LiDAR systems, the category of Wind LiDARs is of special interest. The capability of accurate wind measurements and atmospheric temperature determination is highly significant in the domains of weather prediction models, civil and military aircraft and air transport in general, defence industry, atmosphere dynamics research, as well as climate change surveillance. Wind LiDARs are separated into two types, based on the wind detection technique used.

Direct Motion Detection Wind LiDARs use the definition of velocity (Equation 2.1.1),

$$\vec{V} = \frac{\vec{dr}}{dt} \tag{2.1.1}$$

where V is the wind's velocity, Δr is its change of position and t is time. This method requires the existence of trackers, such as clouds or aerosols, to measure their position shift with time, either in real time or through pictures taken [15].

The second technique exploits the Doppler Effect to determine the wind's velocity, according to Equation 2.1.2. Δw is the detected frequency shift, k is the emitted wave's wavenumber and V is the wind's velocity.

$$\Delta\vec{w} = -2k\vec{V} \tag{2.1.2}$$

2.2 Doppler Effect

The Doppler Effect refers to the observed change in frequency of a wave, when the wave's source and the observer are in relative motion to each other. It is named after the Austrian physicist Christian Doppler, who described the phenomenon in 1842 for acoustic waves. It does, though, also apply to electromagnetic waves and radiation.

In classical physics [20], where the speeds of the source and the receiver relative to the medium are lower than the velocity of waves in the medium, the relationship between the observed frequency f' and the emitted wave's frequency f is given by 2.2.1:

$$f' = (\frac{u \pm u_0}{u \mp u_s})f \tag{2.2.1}$$

,where

- u is the propagation speed of waves in the medium
- u_0 is the speed of the receiver relative to the medium
- u_s is the speed of the source relative to the medium

Using this formula, detecting the frequency shift of the emitted radiation can lead to wind velocity determination. The detection of the Doppler Shift in LiDAR systems can either be direct (Direct Detection Doppler LiDAR) or coherent (Coherent Detection Doppler LiDAR).

In Direct Detection Doppler LiDARs, an optical frequency discriminator or spectrum analyzer is used to convert the Doppler frequency shift to a change in optical intensity or power, which is in turn directly detected.

In Coherent Detection Doppler LiDARs, on the contrary, the return signal is optically mixed with a local oscillator laser, and the resulting beat signal has the frequency (except for a fixed offset) equal to the Doppler shift due to the moving particles.

2.3 Coherent Detection

More specifically, the layout used in a Coherent Detection Wind LiDAR is the following (Figure 2.1):

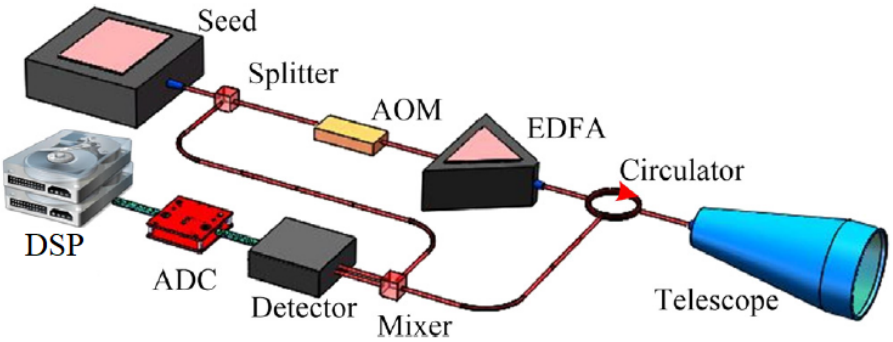


Figure 2.1: The CDWL layout [27]

At the first stage, the seed emits CW (Continous Wave) laser. The beam splitter divides the laser into local oscillator and transmitted laser. The continuous transmitted laser is chopped into a pulse train with a frequency shift of 80 MHz and 300 ns pulse duration by using an acoustic-optical modulator (AOM). The laser is then amplified, going through an erbium doped fiber amplifier (EDFA). In it, a relatively high-powered beam of light is mixed with the input signal using a wavelength selective coupler. The input signal

and the excitation light must be at significantly different wavelengths. The mixed light is guided into a section of fiber with erbium ions included in the core. This high-powered light beam excites the erbium ions to their higher-energy state. When the photons belonging to the signal at a different wavelength from the pump light meet the excited erbium ions, the erbium ions give up some of their energy to the signal and return to their lower-energy state, amplifying the laser with a pulse energy of 100 µJ. Then, the pulse passing through the circulator is collimated and sent to the atmosphere by a telescope. The backscattering photons are collected by the same telescope and finally mixed with local oscillator in the 3 dB coupler (mixer). Then, the mixed signal is measured by the balanced detector and digitalized by an ADC (Analog to Digital Converter) [27].

An analog-to-digital converter (ADC) is a system that converts a continuous-time and continuous-amplitude analog signal to a discrete-time and discrete-amplitude digital signal.

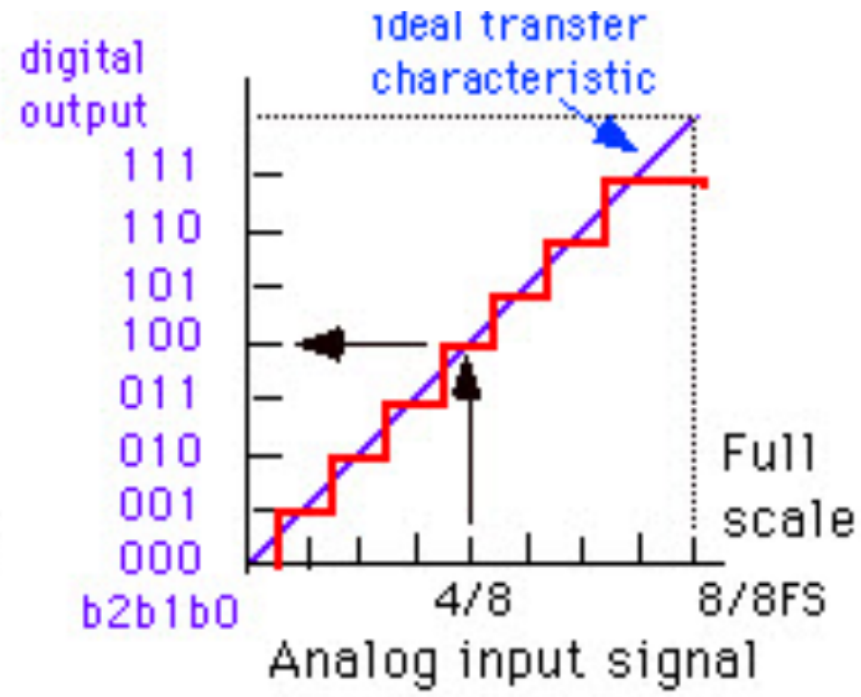


Figure 2.2: ADC operation [5]

Typically the digital output is a two's complement binary number that is proportional to the input. The conversion involves quantization of the input, so it necessarily introduces a small amount of error or noise. Furthermore, instead of continuously performing the conversion, an ADC does the conversion periodically, sampling the input, limiting the allowable bandwidth of the input signal. The ADC samples the analog signal on each falling or rising edge of sample clock. In each cycle, it receives the analog signal, measures and converts it into a digital value. The ADC converts the output data into a series of digital values by approximating the signal with fixed precision. In ADCs two factors

determine the accuracy of the digital value that captures the original analog signal. These are quantization level or bit rate and sampling rate. Figure 2.2 depicts how analog to digital conversion takes place using a 3-bit ADC. Bit rate defines the resolution of digitized output. One of the major benefits is a high data acquisition rate even at multiplexed inputs. With the invention of a wide variety of ADC integrated circuits, data acquisition from various sensors becomes more accurate and faster. Dynamic characteristics of the high performance ADCs are improved measurement repeatability, low power consumption, precise throughput, high linearity, excellent Signal-to-Noise Ratio (SNR) and so on [5].

The raw data is, after being digitized, ready either for processing in real time by digital signal processors (DSP), or to be stored in a disk array for post-processing.

The main application of Wind LiDAR data processing concerns wind velocity computation. This exploits the Doppler Effect, computing the frequency shift between the emitted radiaton and that backscattered by the aerosols and atmospheric particles. The process to achieve this is the following: the laser emitted by the CDWL has a certain frequency f_c. As already explained, the signal scattered back to the system's telescope has its frequency altered by particles, aerosols or gases in the atmosphere, because of the Doppler Effect. If the received signal's frequency is by f_d different from the initial laser's f_c frequency, the mix of the original with the backscattered signal in the system's coupler results in a sine wave, as presented in Figure 2.3.

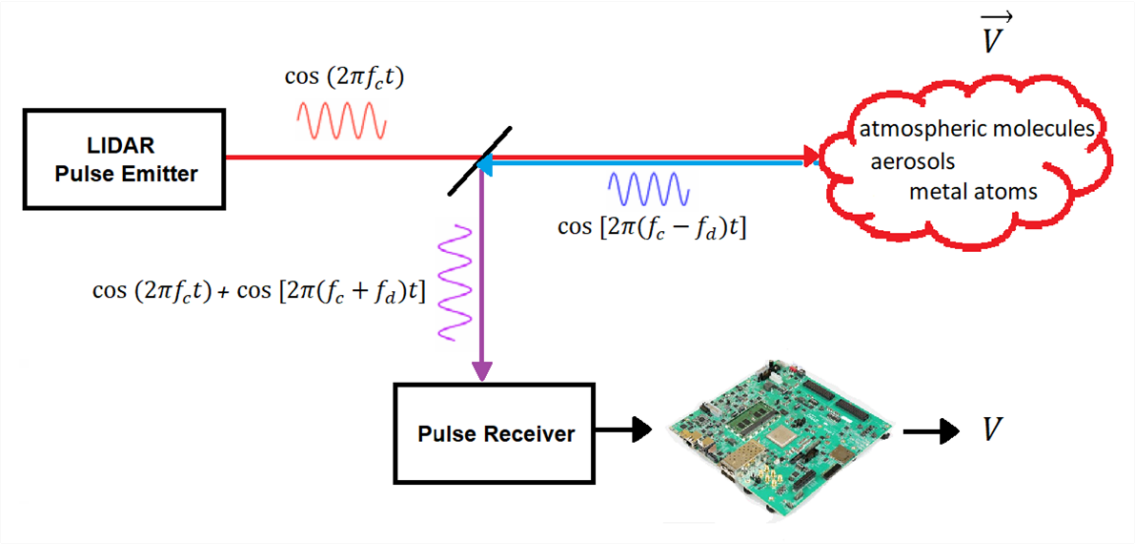


Figure 2.3: CDWL operating principle [15]

A more advanced mathematical approach of the phenomenon can be issued by modeling the emitted $E(t)$ and received signals $r(t)$ as [15]:

$$E(t) = \sqrt{2P}cos[2\pi f_c t + \theta(t)] + L_R(t) \tag{2.3.1}$$

$$r(t) = k\sqrt{2P}\sum_{i=0}^{L-1}\alpha_i cos[2\pi(f_c \pm \Delta f_i)t + \theta(t) + \phi_i] \tag{2.3.2}$$

where

- k is a proportionality constant, ‘
- f_c is the laser (carrier) frequency ,
- $\theta(t)$ is the laser phase noise
- $L_R(t)$ is the laser relative intensity noise (RIN),
- α_i are backscatter coefficients associated with aerosol particles,
- Δf_i is the Doppler shift associated with the movement of the i_{th} particle
- ϕ_i is the phase factor associated with the i_{th} particle

For the sake of simplicity, instead of studying the signals backscattered from all moving targets in the atmosphere, the receive signal by a single target can be adopted, neglecting the effect of phase noise. This does not negate its wider application. The impact of pulsing and not using CW laser should also be taken into consideration. Using a pulse function in the time domain [15]

$$s(t) = u(t) - u(t - T) \tag{2.3.3}$$

a model of the received signal, scattered back by a single target is the following:

$$r(t) = \alpha\sqrt{2P_s}s(t - t_0)cos[2\pi(f_c \pm \Delta f)t + \theta(t) + \phi] \tag{2.3.4}$$

Its transformation into the frequency domain using the Continuous Time Fourier Transform is:

$$R(F) = S(F) \circledast [\frac{\sqrt{2P_s}\alpha}{2}e^{j\phi}\delta(F + f_c + \Delta_f) + \frac{\sqrt{2P_s}\alpha}{2}e^{-j\phi}\delta(F - f_c - \Delta_f)] \tag{2.3.5}$$

with a Power Spectral Density of

$$|R(F)|^2 = [\frac{P_s\alpha^2}{2}[|S(F + f_c + \Delta_f)|^2 + |S(F - f_c - \Delta_f)|^2] \tag{2.3.6}$$

Reviewing the power spectrum of the backscattered signal, its main frequency can be determined detecting the spectrum's peaks.

In the ideal case, the frequency analysis of a single backscattered pulse would be enough for the computation of its frequency and consequently its difference to the laser's frequency for the determination of the wind velocity. This however is not possible, for reasons explained hereupon, in Paragraph 2.5.

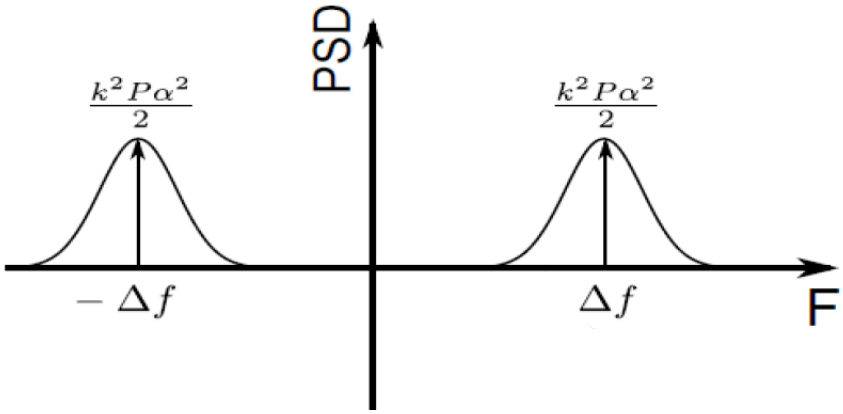


Figure 2.4: Power Spectral Density of signal R(F) [15]

2.4 Frequency Domain Analysis

At first, disregarding the noise, the model previously explained in frequency domain consists of a Gaussian-like distributed coefficient, with its mean μ being the spectrum's central frequency. In reality the frequency power spectrum of the detected backscattered LiDAR signal consists of two main coefficients.

The first one is due to Mie scattering, named after the German physicist Gustav Mie (1868-1957). This coefficient concerns light scattering by aerosols at the lower 4.5 kilometers of the atmosphere, an area containing the Wind LiDAR's detection range. At this height, the emitted signal may encounter round particles of diameters approximately the same to its wavelength.

The second coefficient is a result of molecular scattering, also known as Rayleigh scattering, in honor of the British physicist Lord Rayleigh (1842-1919). In this case, the emitted signal is elastically scattered by molecules of diameters significantly smaller than its wavelength. The backscattered signal's intensity is now inversely proportional to the fourth power of the wavelength.

A depiction of the Frequency Power Spectrum of the ideally detected signal by the layout's telescope is given in Figure 2.5.

The lower intensity coefficient present at the Spectrum is the Rayleigh one. It is obvious that, as expected by Equation 2.3.6, this coefficient is normally distributed, with a mean μ equal to the backscattered signal's frequency.

The Mie coefficient is higher in intensity and less in deviation, also centered around the detected signal's main frequency. It is, however, worth mentioning that the most interesting, in the aspect of research, coefficient is the molecular one, as the atmospheric particles move faster than particles of larger diameters.

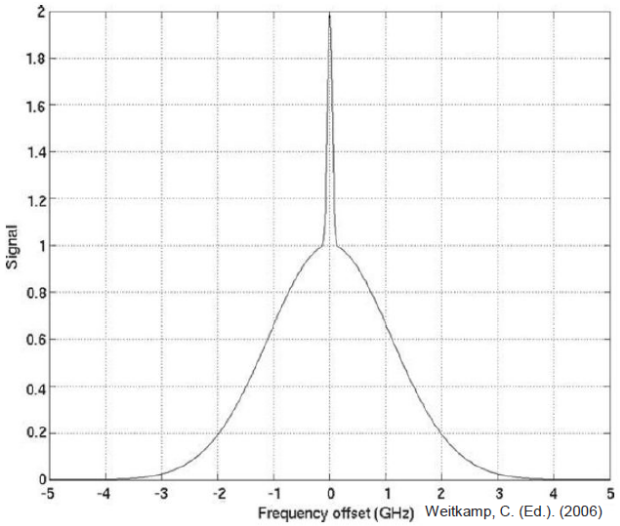


Figure 2.5: Low noise input frequency spectrum [23]

2.5 Challenge: Signal Analysis with extremely low SNR

Even with the existence of these two coefficients, it still seems easy to derive the central frequency of the backscattered radiation from the created frequency spectrum shown in Figure 2.5, either via simple observation, or using an automated digital procedure. This, however, is not possible, because this spectrum in real conditions contains high levels of noise for various reasons [15]. Realistically, such a measurement might even include noise of SNR (Signal to Noise Ratio) down to -20dB.

The first and main problem is the low intensity of the backscattered radiation. Both particle and aerosol scattering in the atmosphere are considered to be isotropic [13]. This means that, in both cases, photons are allowed to be scattered at any direction of the 3-D space. It is, therefore, simple to understand that the amount of photons scattered in the exact opposite - or at least almost opposite - to their initial direction is significantly smaller than the number of the photons of the radiation initially emitted. This can, qualitatively, be translated as low intensity of useful data in the detected signal, based on the LiDAR laser's wavelength.

Another important noise source for the radiation received is the solar radiation. The solar radiation backscattering's misleading contribution to Wind LiDAR measurements during the day is huge, as a result of its high intensity and its coefficients' variety in wavelengths. The sunlight-induced noise can be of orders higher than weak LiDAR return signals from bright surfaces such as landmass, snow and -in this case- thick clouds [26]. For optically thin atmospheric constituents like aerosols and super-thin clouds, sunlight contamination can also introduce large uncertainty in LiDAR measurements. The sunlight's effect on the received signal is on display on the measurements of Figure 2.6, concerning the mean average and standard deviation of LiDAR measurements' noise during day and night. It is obvious that even the highest night noise levels only reach the lowest of the

equivalent day levels.

Except for the mentioned external noise sources to the Wind LiDAR measurements, the LiDAR system does also import some noise coefficients to its measurements itself, as do all mechanical systems. The first one is a shot noise coefficient, due to the layout's oscillator, following the Poisson distribution. In this case, the oscillator shot noise is a result of the discrete nature of the photon absorption process, which leads to fluctuations of the number of photons detected (or simply counted in the abstract). The thermal noise present in every conductor, and therefore in all conductors used in the CDWL, is a noise source, too.

Finally, another barrier - if not cause of noise to the measurements - is the differentiation of scattering properties of different particles in the atmosphere. This fact can be exploited in LiDAR applications concerning atmospheric particle classification. In wind velocity measurements, however, this can lead to different spectral coefficients, some of which contain more important information than others as explained in Paragraph 2.4 and so some kind of coefficient separation has to take place.

2.6 Related Research and Implementations

Many researchers have contributed to the development and improvement of wind measurements using Coherent Detection Doppler LiDAR. Their work concerns areas such as improving the LiDAR system, modeling and mathematically analyzing the signals of the wind velocity detection process. Other works concern noise profiling or investigating noise handling techniques. The transformation of the received signal into the frequency domain and its accuracy are other crucial topics that have also been touched upon.

Data processing begins from data modeling. Optimizing the design of any system stipulates a prior expectation of its response to different inputs and conditions. *Coherent Doppler Lidar Signal Spectrum with Wind Turbulence* [19], as well as Professor Xinzhao Chu's lectures at the University of Colorado Boulder [15] include the mathematical analysis of the signals in the procedure. These time domain models apply to the frequency domain, too. Frehlich and Cornman [19] did also model the SNR of the system, as well as the expected error, specifying all variables affecting them, enabling theoretical optimizations.

The wind measurement system is, certainly, not the same in every differentiation of Doppler Wind LiDAR. The processing procedure always has to fit to the way data are collected by the LiDAR telescope, and Doppler LiDAR evolution requires evolution and adaptation of this procedure. Report *Remote Sensing of Complex Flows by Doppler Wind Lidar: Issues and Preliminary Recommendations* [16] keeps track of the state of the art Wind LiDAR systems and use cases, their data needs, method of computing wind velocity and finally of its pros and cons. This can inspire further research into the techniques and lead to a choice depending on the application of interest.

The credibility of those measurements is dependent on the system's capability of reducing input noise. Most Wind LiDAR systems average the results of even 10-minute

scans to reach satisfyingly low error. Some propositions [28] include the total rejection of samples received from SNR levels lower than a specific value. A common technique [28, 25] concerns spectral accumulation and averaging to reduce the White Gaussian Noise. Other propositions [21] include further filtering, by spectral whitening and thresholding, as well as removing low frequency noise.

This problem also contains digital processing interest, as the algorithm used to transform the signal and accumulate the spectra can affect both measurement accuracy and execution time. This processing usually concerns high level programming and CPU processing for the sake of simplicity and development ease. Time to frequency domain transformation usually occurs using digital forms of the Fourier Transform. The most common digital algorithm used is the Fast Fourier Transform (FFT) [28, 21], while the simple Digital Fourier Transform is also used in some cases [25]. Modern approaches also propose the usage of a Joint Time-Frequency Analysis method instead [27], possibly leading to even clearer results

While the measurements' accuracy is non-negotiable, the systems can be further optimized using application specific designs with a matching resources distribution and exploitation. This cannot be achieved by general purpose CPUs.

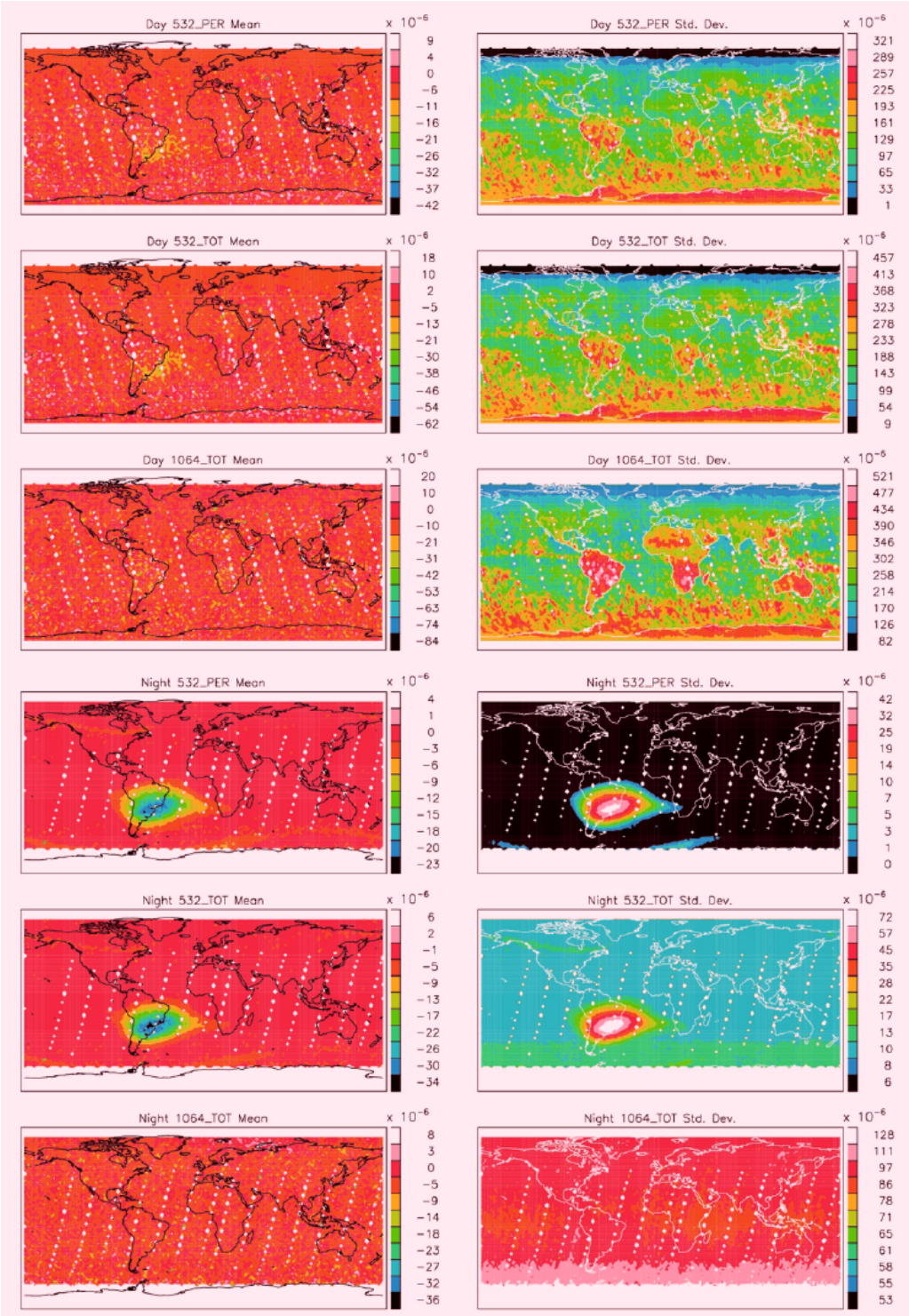


Figure 2.6: Mean average (μ) and standard deviation (σ) of sunlight noise during day and night [29]

Chapter 3

SoC FPGA

3.1 FPGA Fundamentals

A field-programmable gate array (FPGA) is a semiconductor integrated circuit designed to be configured by a customer or a designer after manufacturing, hence the term "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC).

FPGAs contain an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects that allow the blocks to be wired together. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

Unlike processors, FPGAs are truly parallel in nature, so different processing operations do not have to compete for the same resources. Each independent processing task is assigned to a dedicated section of the chip and can function autonomously without any influence from other logic blocks. As a result, the performance of one part of the application is not affected when more processing is added.

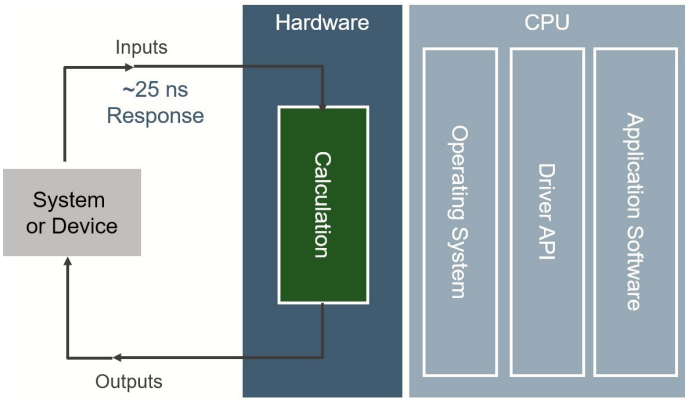


Figure 3.1: FPGA parallel to CPU

3.2 FPGA Architecture

FPGA chips are made up of a finite number of predefined resources with programmable interconnects to implement a reconfigurable digital circuit and I/O blocks to allow the circuit to access the outside world. These resources can be separated into Logic Blocks and Hard Blocks [4].

3.2.1 Configurable Logic Blocks

The configurable logic blocks (CLBs) are the basic logic unit of an FPGA. Sometimes referred to as slices or logic cells, CLBs are made up of two basic components: flip-flops and lookup tables (LUTs). Various FPGA families differ in the way flip-flops and LUTs are packaged together.

Flip-flops are binary shift registers used to synchronize logic and save logical states between clock cycles within an FPGA circuit. On every clock edge, a flip-flop latches the 1 or 0 (TRUE or FALSE) value on its input and holds that value constant until the next clock edge.

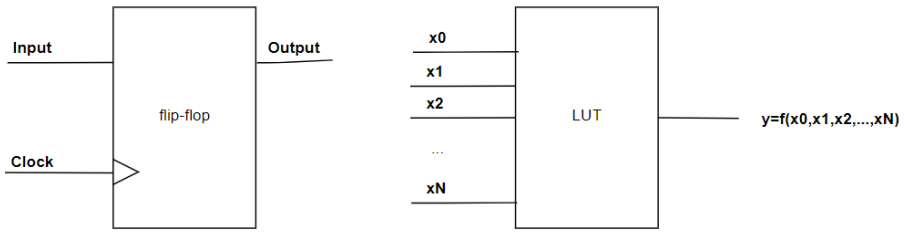


Figure 3.2: The flip-flop and LUT symbols

Much of the logic in a CLB is implemented using very small amounts of RAM in the form of LUTs. It is easy to assume that the number of system gates in an FPGA refers to the number of NAND gates and NOR gates in a particular chip. But, in reality, all combinatorial logic (ANDs, ORs, NANDs, XORs, and so on) is implemented as truth tables within LUT memory. A truth table is a predefined list of outputs for every combination of inputs.

3.2.2 Hard Blocks

Modern FPGA families expand upon the above capabilities to include higher level functionality fixed in silicon. Having these common functions embedded in the circuit reduces the area required and gives those functions increased speed compared to building them from logical primitives. Examples of these include multipliers, generic DSP blocks and embedded memories.

The seemingly simple task of multiplying two numbers together can get extremely resource intensive and complex to implement in digital circuitry. Indicatively, multiplying

two 32-bit numbers together could even take more than 2000 operations . Because of this, FPGAs have prebuilt multiplier circuitry to save on LUT and flip-flop usage in math and signal processing applications.

Higher-performance FPGAs have prebuilt multiplier-accumulate circuitry. These prebuilt processing blocks, also known as DSP48 slices, integrate a 25-bit by 18-bit multiplier with adder circuitry.

Memory resources are another key specification to consider when selecting FPGAs. User-defined RAM, embedded throughout the FPGA chip, is useful for storing data sets or passing values between parallel tasks. Depending on the FPGA family, onboard RAM may be configured in blocks of 16 or 36 kb. Block RAM is highly significant, because the option to implement data sets as arrays using flip-flops is really expensive for FPGA logic resources. Digital signal processing algorithms often need to keep track of an entire block of data, or the coefficients of a complex equation, and without onboard memory, many processing functions do not fit within the configurable logic of an FPGA chip.

3.3 Benefits and Applications

FPGA usage is ideal for many digital processing systems, offering the hardware developer a variety of benefits [3, 6]:

- **Acceleration:** Although FPGAs do generally operate at lower frequencies than other processing units (e.g. CPUs), their architecture is application specified, allowing parallelism and the usage of resources in a way that processing operations do not compete one another. These result to significant improvement of execution time. FPGAs can also be bought "off the shelf" vs. e.g ASICs, which require manufacturing cycles taking many months and FPGA designs can be sold right after their operation is tested

- **Flexibility:** FPGA functionality can change upon every power-up of the device. An architectural change is available to be tested just by downloading a new configuration file into the device. This means that simple FPGA changes can replace needs of costly PC board changes.

- **Integration:** Today's FPGAs include on-die processors, fast transceiver I/O's, RAM blocks, DSP engines and more. More functions within the FPGA mean fewer devices on the circuit board, increasing reliability by reducing the number of device failures.

- **Total Cost of Ownership:** While ASICs may cost less per unit than an equivalent FPGA, building them requires a non-recurring expense (NRE), expensive software tools, specialized design teams, and long manufacturing cycles. FPGAs usually support long lifecycles (15-years or more), avoiding the cost of redesigning and requalifying production equipment if one of the electronic devices on-board goes end

of life. They also reduce risk, allowing prototype systems to ship to customers for field trials, while still providing the ability to make changes quickly before ramping to volume production.

All the FPGA benefits mentioned above have led to their usage in a wide variety of application areas. Some of them concern electronics, either consumer (digital displays and cameras, printers) or scientific (instruments such as lock-in amplifiers, boxcar averagers), communications, both wired (optical transport networks, connectivity interfaces) and wireless (mobile backhaul, baseband), automotive (vehicle networking and connectivity, automotive infotainment, image processing) and aerospace and defence (Avionics, communications, missiles, space). FPGA systems also apply to medical (ultrasound, X-ray, surgical systems) and financial areas (crypto mining, high-frequency trading) and so on and so forth. Of primary importance, though, is their contribution to high performance computing, in servers, super computers, high-end RADAR and Beam Forming systems and in integrated systems design, being the perfect subject for ASIC prototyping and computer hardware emulation.

3.4 System on a Chip

A System on a Chip (SoC) is a complex IC that integrates digital, analog, radio frequency components (i.e. processor, hardware accelerators, ADCs/DACs, memory, interfaces and interconnection, peripherals etc.) all in a single chip. As they are integrated on a single substrate, SoCs consume much less power and take up much less area than multi-chip designs with equivalent functionality. Because of this, SoCs are very common in the mobile computing (such as in smartphones) and edge computing markets. Systems-on-chip are typically fabricated using metal–oxide–semiconductor (MOS) technology, and are commonly used in embedded systems and the Internet of Things. Except for the low power consumption and the size reduction, SoCs also offer increased performance, reliability, as well as reduction of the system's cost.

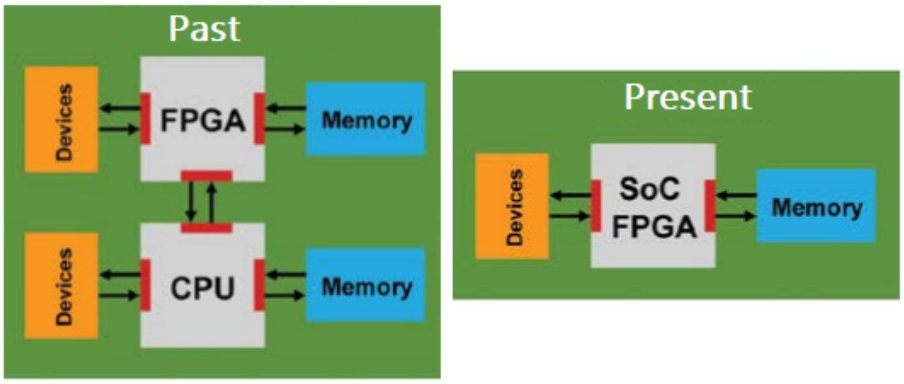


Figure 3.3: CPU + FPGA vs SoC FPGA [7]

Processors and FPGAs are the hardworking cores of most embedded systems. Integrating the high-level management functionality of processors and the stringent, real-time operations, extreme data processing or interface functions of an FPGA into a single device all constitute an even more powerful embedded computing platform. SoC FPGA devices integrate both architectures into a single device. Consequently, they provide higher integration, lower power, smaller board size, efficient HW/SW co-processing and higher bandwidth communication between the processor and FPGA. They also include a rich set of peripherals, on-chip memory, an FPGA-style logic array and high speed transceivers.

At present, the sets of SoC FPGAs available on the market are mainly ARM based and fabricated by Xilinx (Zynq 7000 AP SoC, Zynq UltraScale+ MPSoC), Intel (Stratix 10 SoC, Arria 10 SoC, Arria V SoC, Cyclone V SoC) and Microsemi (SmartFusion SoC, SmartFusion 2 SoC) [6].

3.5 FPGA in LiDAR Applications

All FPGA and SoC benefits have led to the spreading of this computation means and its usage in many fields. LiDAR applications are no exception. FPGA and LiDAR are combined in fields such as land transport, aircraft, atmosphere research and weather prediction.

FPGA designs for LiDAR usage are not restricted to the implementation of digital processing algorithms only. Chevrier and Campanella [14], for instance, propose the usage of an FPGA based Time to Digital Converter (TDC) to digitize the received signal instead of a conventional ADC used in the majority of applications. This kind of sampling is triggered at level crossing and not based on clock input as do ADCs. It is suggested that the reward for this is improved noise immunity.

LiDAR acquired data are not only processed in real time, but they can also be stored for post processing and research. The large volume of data, though, must somehow be dealt with. A hardware - FPGA based compression system proposed in *Hardware accelerated compression of LIDAR data using FPGA devices* [12] can significantly improve the speed of the procedure compared to software compression. Another benefit is processor load alleviation.

The most common FPGA area of application is probably digital signal processing acceleration. In *Real-Time LiDAR Processing for Drivable Region Segmentation on an FPGA* it is proposed that hardware implementation of computer vision algorithms in collaboration with convolutional neural networks (CNN) [22] may lead to further development of autonomous vehicles, via drivable region segmentation.

All benefits considered, the embedding of SoC FPGA platforms on Doppler Wind LiDAR systems is not a new idea. NASA have researched the embedding of powerful FPGA boards on LiDAR based scientific instruments (HOPS project) [11] to remove time-critical, intensive computations from CPU load. The scalability and reconfiguration options lead to ease in the synergy of HOPS with various NASA missions, including 3-D

Winds missions.

Coherent Doppler LiDAR data processing on FPGA platforms was theoretically established more than ten years ago. *Recent Advances in Signal Processing* [34] includes propositions of data processing design, FPGA platform and wind velocity determination. Based on that, more prototypes [8] have been developed, with different LiDAR systems and FPGA platforms used for real time signal data processing.

Optimizations of these systems require research and experimenting, and can be achieved in the fields of acceleration, power consumption and accuracy.

Chapter 4

Developed Processing System

4.1 Proposed System Architecture

As explained in Chapter 2, a Wind LiDAR system's measurements may be highly noisy, especially in a long distance from the telescope and unfavourable atmospheric conditions context. In several circumstances, the noise levels could even reach SNR (Signal to Noise Ratio) levels of -20dB, rendering the received signal from a single pulse highly unreliable. The requirement for high precision in the wind velocity computation generates the necessity of handling this noise.

Except for the signal processing, a systemic approach to the design leads either to the development of more circuitry or to the utilization of non-FPGA Soc resources for control of the LiDAR position and scanning process and interaction with the user.

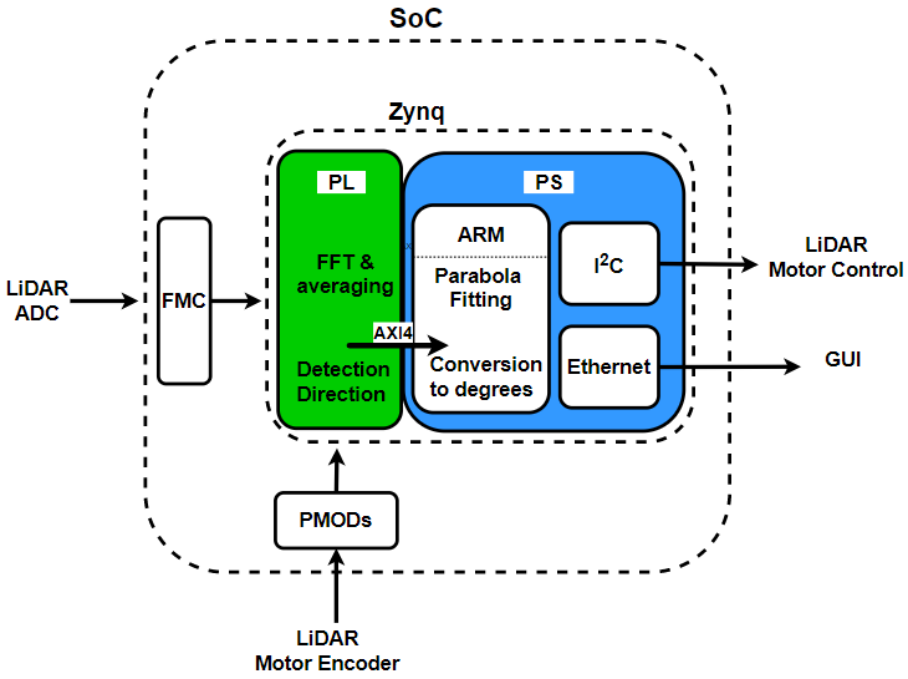


Figure 4.1: Proposed Design

Figure 4.1 depicts all the parts of the system proposed. For this design it is assumed that the sampling rate of the ADC equals to the operation frequency of data processing by the FPGA, set to be 500 MHz. The FPGA part will be responsible for the hard computing task of time to frequency domain transformation, using the Fast Fourier Transform algorithm [17] and the accumulations required for the averaging process.

It is also important to be able to determine the azimuth and altitude of the emitted radiation, so as to specifically determine the atmospheric volume for which the wind velocity is computed. Azimuth and altitude are angular measurements in a spherical coordinate system. The vector from an observer (origin) to a point of interest is projected perpendicularly onto a reference plane; the angle between the projected vector and a reference vector on the reference plane is called the azimuth. The angle between the point and the observer's local horizon is called the altitude or elevation angle (Figure 4.2).

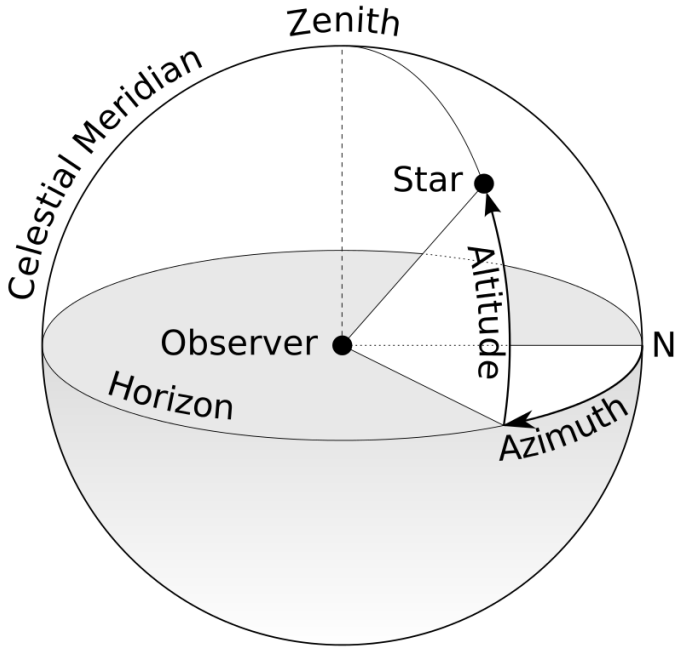


Figure 4.2: Azimuth and Altitude [2]

In this direction, another PL circuit will be in charge of counting the transitions of two signals produced by the LiDAR azimuth and elevation motor encoders.

The number of pulses recorded will then be translated into angle of rotation through a simple C code using the ARM of the SoC. The PS of the system will also execute the final computation of wind velocity, using the parabola fitting technique on the FPGA output data.

Curve fitting is the process of constructing a curve, or mathematical function, that has the best fit to a series of data points. When a 2^{nd} grade polynomial is used, then the method is called parabola fitting.

Suppose A,B,C are the consecutive points mentioned, with B being the maximum, A being the point corresponding to the previous spectral frequency and C to the next

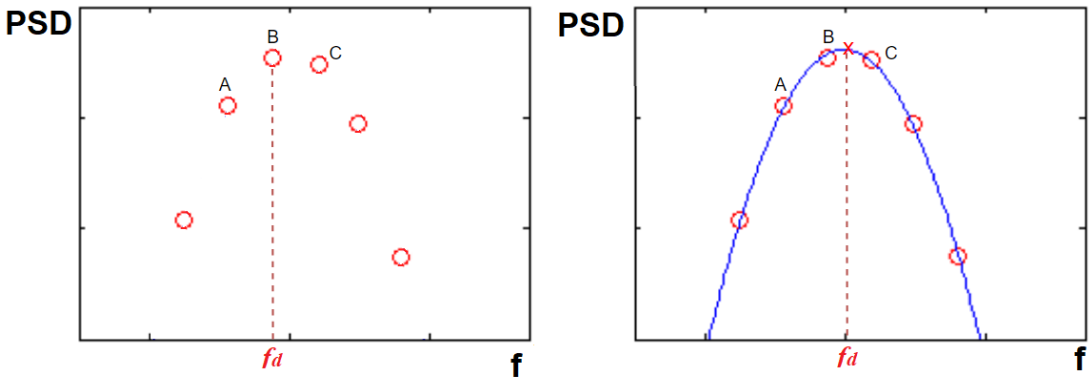


Figure 4.3: Frequency detection without (left) and with (right) parabola fitting

(Figure 4.3). Differentiating the parabola which fits these three points, the maximum can be found at the the derivative's zeroing frequency $f_d = f_B + \Delta f$ [34], where

$$\Delta f = \frac{f_A - f_C}{2(f_A + f_C - 2f_B)} \tag{4.1.1}$$

resulting to

$$f_d = f_B + \frac{f_A - f_C}{2(f_A + f_C - 2f_B)}. \tag{4.1.2}$$

The SoC's IIC connector can be used to exchange messages with the LiDAR, using the IIC protocol [9] and fixed functions of an I2C software library, determining its situation and controlling its rotation and scanning. Finally, a GUI can be developed and communicate with the embedded on the LiDAR SoC using the platform's Ethernet connector, for remote control and communication.

In the limits of this thesis, the two key components of the design which were developed and tested on the FPGA are the FFT and averaging design and the angle counter. On the PS side, C codes to execute the counter to angle transformation and the parabola fitting computation of Equation 4.1.2 were also written and used.

4.2 Noise Compensation Method: Averaging

For noise compensation, research of the expected signal is required, after adding noise to the already known spectral coefficients, Mie and Rayleigh of Figure 2.5. Noise modeling has been based on the fact that coherently detected signals are modeled as narrowband rf signals with additive white Gaussian noise (AWGN) [10]. This noise model follows the normal distribution, rendering all its discrete points independent of each other and randomly distributed.

This stochastic nature of the system's measurement noise has been used to reduce it. Implementing the model in MATLAB and receiving the Power Spectrum of a single pulse detected, adding AWGN of -20dB leads to the expected noisy output of Figure 4.4. In

this specific simulation, the signal's frequency to be detected was 80MHz, a frequency impossible to be selected as this Figure's highest peak.

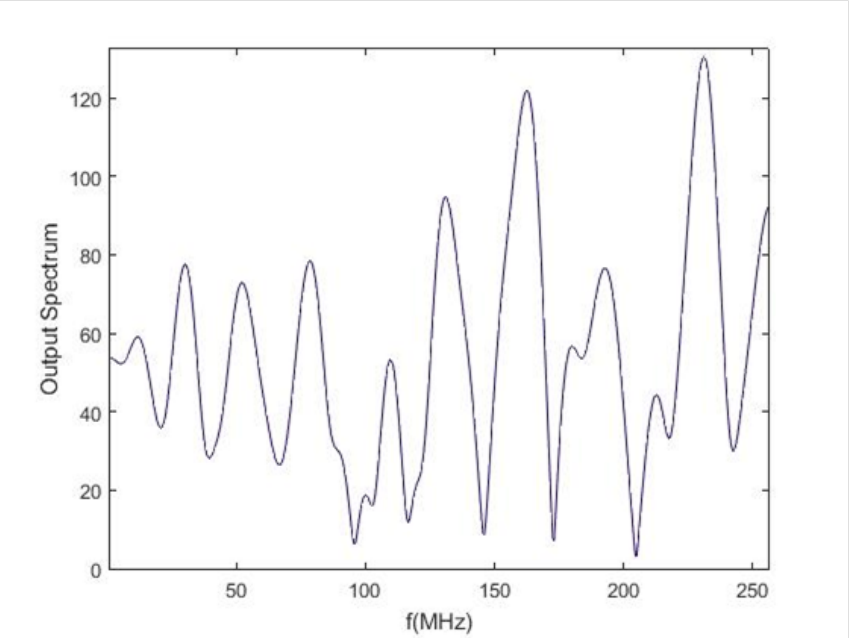


Figure 4.4: Power Spectrum for a single backscattered pulse

It is clear that the output of Figure 4.4 is far from the ideal Power Spectrum depicted in Figure 2.5. Knowing, however, that the noise in these measurements is normally distributed, repeating the procedure enough times and accumulating the individual spectra can lead to the solution of the problem. Actually, all individual spectra consist of one important PSD peak corresponding to the backscattered signals' frequency and many noise-generated peaks in different frequencies with various density values. Accumulating such spectra, the only common peak in all of them is the one at the frequency to be detected. This procedure gradually leads to a spectrum with one clear peak and an offset of noise as shown in Figure 4.5. From the spectrum created, it is now possible to derive the desired frequency f_d, and thus compute the wind velocity.

Overall, wind velocity determination requires the recording, storage and accumulation of many successive pulses. The optimum implementation of a digital system to fulfill those requirements presupposes the perfect understanding of the procedure of recording those pulses by the Wind LiDAR.

The system under research is pulsed, meaning that the laser emitted is not continuous, but separated into pulses. The LiDAR spatial resolution is given by Equation 4.2.1 [27]:

$$SR = \frac{\Delta T * c}{2} \tag{4.2.1}$$

where

- ΔT is the LiDAR's pulse duration and

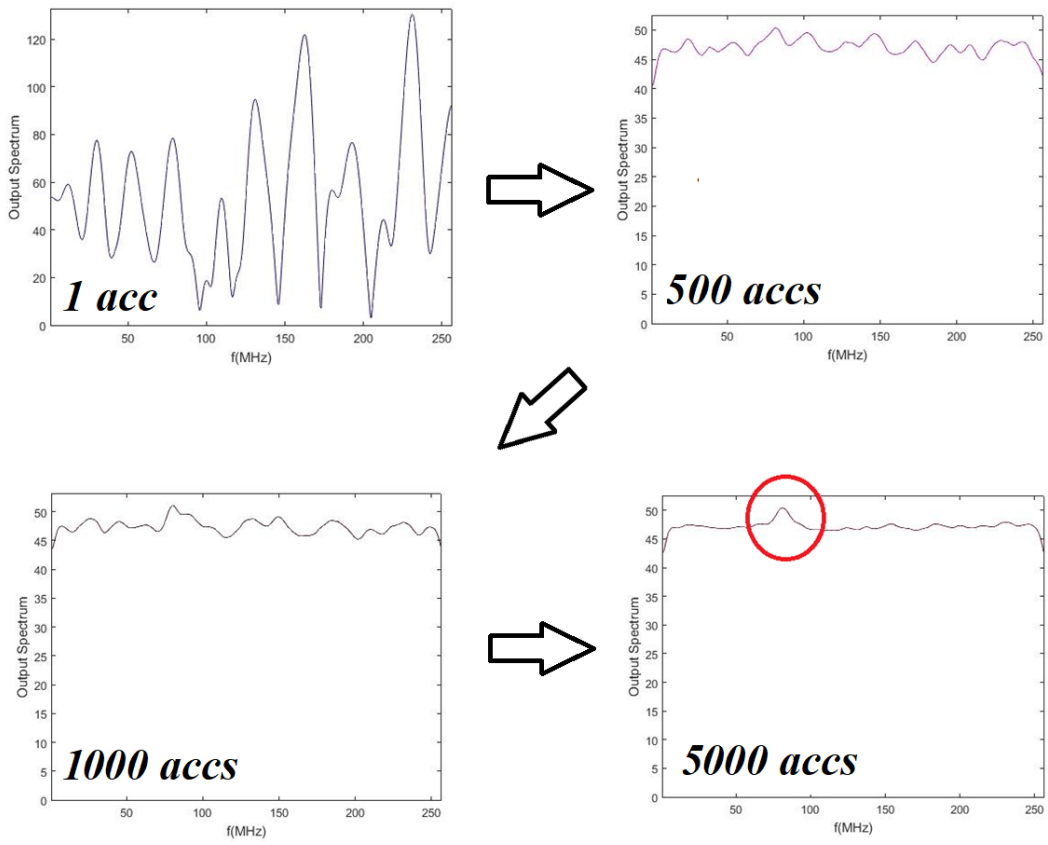


Figure 4.5: Noise elimination through spectral accumulation

- c is the speed of light in the atmosphere

Equation 4.2.1 mathematically implies that increasing the pulse duration will also result to an increase of uncertainty of the area of detection. The atmosphere is now to be parted into pieces, or else gates, in each one of which the wind velocity is going to be computed. Atmospheric partition has already been used in models for the research of LiDAR data properties (e.g Feuilleté model [24]). For each gate in the atmosphere a number of samples is going to be recorded, from now on referred to as gate length L_g. The need for existence of $L_g > 1$ samples is given, as the more samples recorded for each pulse, the more information available and so the easier the computation of wind velocity for this certain area. That, however, leads to a tradeoff, because increasing the gate length L_g also increases the uncertainty of the area for which the wind is measured.

The LiDAR initially emits a pulse and after a specific period of time, the system's telescope receives the first backscattered photons. Recording the first L_g samples is actually recording the first backscattered pulse to be processed. This data is considered to originate from the closest to the telescope gate in the atmosphere, in the LiDAR's radial direction. The following L_g samples correspond to the second gate, the next L_g to the third and so on, till the samples from the edge of the LiDAR's radius of detection are

received.

Having saved data for each consecutive gate separately, a frequency domain transformation using the Fast Fourier Transform results to each point's first power spectrum, which is still spoiled with noise. The LiDAR may now emit a second pulse, which will consequently result to L_g backscattered samples for each gate. Transforming into spectra and accumulating with the already saved corresponding ones leads, after many accumulations, to the detection of the actual input signal's frequency for each gate of the atmosphere.

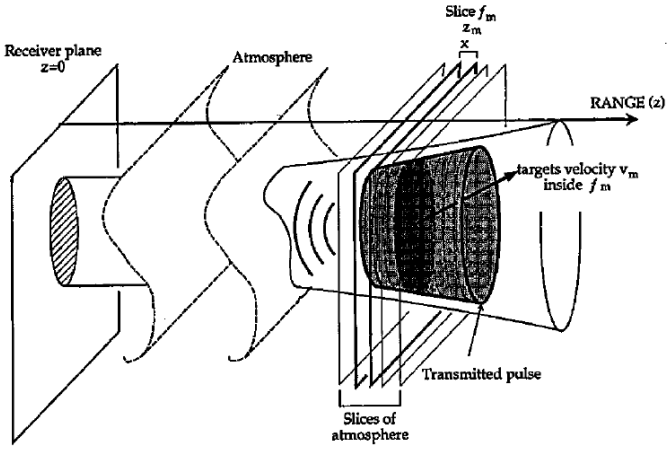


Figure 4.6: Feuilleté Model [24]

Each gate consists of L_g points, having therefore a real length of $l = L_g * c * T_{ADC}$. Having determined the Radius of Detection and supposing an ideal pulse (zero width), a simple division can lead to the number of gates, for which the processing system can compute the wind velocity.

$$N_p = \frac{R_D}{l} \tag{4.2.2}$$

where

- N_p is the number of gates,
- R_D is the LiDAR's detection radius,
- l is each atmospheric gate's actual length

The correct and efficient operation of the data processing system does consequently require:

I A choice of gate length L_g to satisfy the required velocity detection precision, while also remaining rather small in order to achieve clear spatial resolution.

II A choice of FFT length to satisfy the required velocity detection precision, with the minimum length possible to spare computing load (resources, time, power).

III A choice of SoC platform to satisfy the design needs in memory resources, while also supporting the highest possible processing frequency to minimize the execution time.

IV Efficient design in HDL level, with parallel, pipelined function and the optimum utilization of the SoC provided resources.

For requirements (I) and (II) to be met, a MATLAB based variable optimization exploration was carried to be presented in Chapter 5. Using the optimum variables emerging from this exploration, a VHDL design of the data processing system was implemented, also considering the requirements (III) and (IV).

4.3 Proposed System Implementation

4.3.1 Frequency Analysis and Averaging

The PL part of the system will be fed by LiDAR detected data digitized by an ADC of 500MHz sampling frequency, as described in Paragraph 2.3. The FPGA input signal is complex, consisting of a magnitude and an angle coefficient, each 14 bits wide. This data is transformed to the frequency domain with the Fast Fourier Transform algorithm, using the Xilinx Fast Fourier Transform v9.0 LogiCORE IP [31]. This component produces two output signals, among others, corresponding to the real and imaginary parts of each spectral point. These signals are squared using pipelined multipliers and then added to each other. The addition's result is given to Xilinx's CORDIC v6.0 LogiCORE IP [30] for its square root to be computed, a number that equals the magnitude of each spectral point.

Part of the SoC's Block RAM will be used to store the spectra data to be accumulated for each gate of the system. A pipelined adder will also be used to perform this accumulation. Its inputs will be the output of CORDIC IP and the corresponding point of the corresponding spectrum saved in Block Memory at each clock cycle. Its output will overwrite the same point in the RAM Block.

It is, of course, not enough for the design to be operative; its smart and effective function is also requested. It is important that data processing is executed at each and every clock cycle and that all components are processing whenever there is data to be processed. This requires all components constituting the design to operate fully pipelined.

An FFT component is only given L_g samples from each gate to produce a spectrum of N_{FFT} points, and thus the difference is covered by padding $N_{FFT} - L_g$ input samples with zeros. It has practically been researched and proved by MATLAB experiments that detecting the main frequency of the noisy signal, using the process explained, has the same results wherever the zeros are padded. This means that the result is the same for a signal of N_{FFT} points with L_g non-zero samples at the beginning, the end, or anywhere in between.

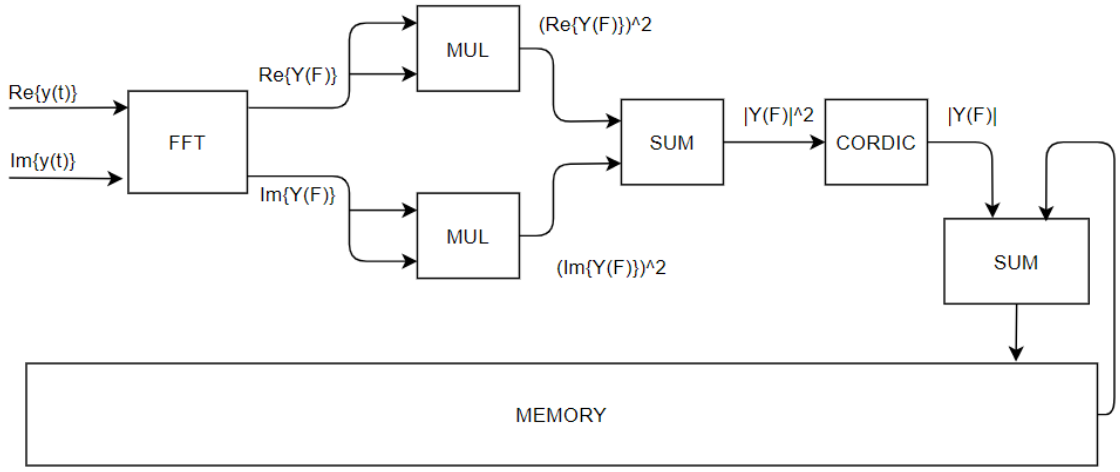


Figure 4.7: Data processing chain

The result of zero-padding, however, is that if a single processing chain is used, the $N_{FFT} - L_g$ input samples following the first L_g should either somehow be stalled to be processed later, creating a problem for the rest of the input stream, or even be ignored. These are no solutions for a real-time application.

To make the optimum use of the system's sampling rate, the choice of L_g should be a perfect divisor of N_{FFT} length and a total of

$$C = \frac{N_{FFT}}{L_g} \tag{4.3.1}$$

processing chains should be used. This way, the system's arbiter component can feed each input sample d_i to the $((d_i \div L_g) \bmod N_{FFT})$ processing chain, feeding the rest $(C-1)$ processing chains with zeros, exploiting the zero-padding placement independence mentioned before. Using C processing chains, C different FFT components can be fed with data at each clock cycle, produce outputs to be further processed by the rest of the design's pipelined components and finally end up with a system's throughput of C instead of 1.

Figure 4.8 presents the input samples distribution to the system's processing chains through time. The LiDAR telescope initially emits a pulsed sine wave and after a certain amount of time the first backscattered radiation sample is detected. This, along with the following $L_g - 1$, will be given, after being digitized by the ADC, as an input to the first processing chain. Those samples originate from the first gate examined. At these clock cycles, the rest of the processing chains will be given a 14-bit vector of zeros as an input. The following L_g samples detected will be fed to the second processing unit and correspond to the second gate, the next L_g to the third and so on, till the first N_{FFT} input samples have been recorded. At the next clock cycle, each of the C FFT components of the design are given N_{FFT} samples, enough for them to compute a whole power spectrum each. Operating pipelined, they are all ready to receive new samples from another gate. It is obvious that the following L_g samples will be fed to the first processing chain once

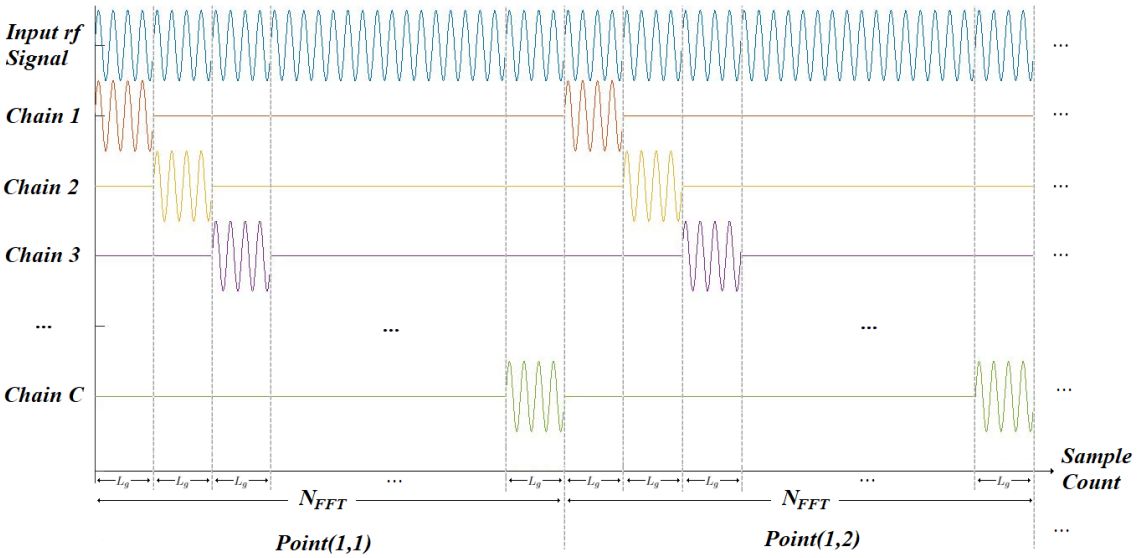


Figure 4.8: Input data routing to processing units

more, originating this time from the $(C+1)^{th}$ gate. So arises the need for each processing chain to contain memory to allow the accumulation of not just one spectrum, but spectra for many gates. A separation of the required memory space to the available processing chains is possible, having determined the number of gates N_P from Equation 4.2.2. For C processing chains available, each one needs space for the storage of N_P/C (or $N_P/C+1$ if the division is not perfect) spectra. Each gate G_i corresponds to the $G_i \div C$ spectrum of the G_i mod C processing chain. Spectral mapping to the processing chain's memory resources is to be seen in Figure 4.9.

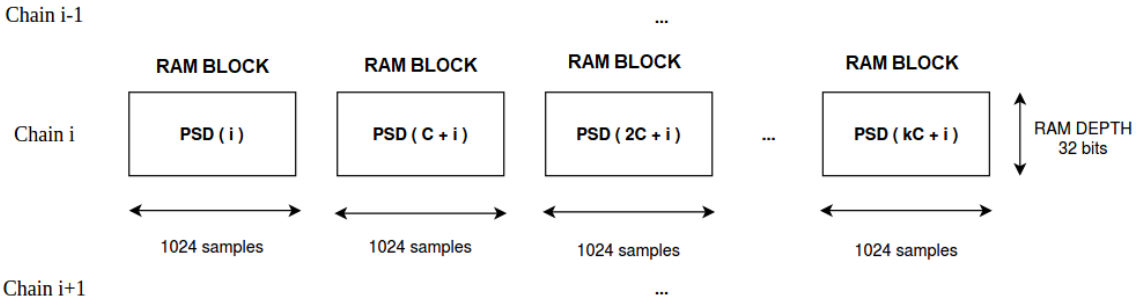


Figure 4.9: PSD mapping to system's memory resources

The design described above is now to be implemented in an HDL language to be tested on FPGA platforms, in this case on a Zynq Ultrascale 102+ platform 4.4. It mainly consists of a VHDL adaptation of Figure 4.7. Its main components are:

- **fft_wrapper:** Xilinx Fast Fourier Transform v9.0 LogiCORE IP. A Xilinx created component for the quick and efficient implementation of the digital Fast Fourier Transform algorithm. A fully pipelined implementation option is chosen for the optimum utilization of the FPGA resources and the maximum data processing rate.

It is, practically, used to convert the input signal from time domain to its frequency domain. Selected input width is 14 bits, same as the selected output width. All configuration information of the FFT IP used in the design is presented in Table 4.1.

data_format	fixed_point
implementation_options	pipelined_streaming_io
input_width	14
memory_options	block_ram
output_ordering	bit_reversed_order
phase_factor_width	14
rounding_modes	truncation
scaling_options	scaled
target_clock_frequency	500
thottle_scheme	realtime
transform_length	1024

Table 4.1: FFT IP Configuration

- **mul14:** Multiplier of two 14-bit inputs, with clock and enable inputs. Xilinx custom component with output width of 28 bits and latency of 4 clock cycles. Used to compute the square of the Real and Imaginary parts of the FFT component outputs.

- **add28:** Adder of 2 28-bit inputs, with clock and enable inputs. Xilinx custom component with output width of 29 bits and latency of 3 clock cycles. Used to add up the outputs of *mul14* component.

- **sqrt:** Xilinx CORDIC v6.0 LogiCORE IP. A Xilinx created component with the option of computing the square root of its input. Input width is set to 32 bits, while output width to 16 bits. The *add28* output's bits are padded with "00" as MSB to be given as input. The output's 15 LSB hold the value of the FFT output's magnitude.

- **dp_ram:** VHDL code for usage of the SoC's RAM Blocks. Generic, dual-port, with optional output register.

- **pip_ram:** Component that uses *dp_ram*, adding pipeline stages in both its input and output, in order to achieve higher operational frequency for the design. Operation experiments of Chapter 5 were carried out with 5 pipeline stages before and 5 after Block RAM Blocks.

- **big_adder:** Adder of a 32-bit and a 15-bit input, with clock and enable inputs. Xilinx custom component with output width of 32 bits and latency of 3 clock cycles.

Used to accumulate the magnitude of each FFT-produced point with its matching point already accumulated and saved in Block RAM. Its output overwrites the matching point in Block RAM.

- **nofft_slice:** Coordinating component of all the previously mentioned components, apart from *fft_wrapper*. In charge of the processing chain's synchronization and the *big_adder*'s inputs' and output's routing from and to the Block RAM.

- **slice:** Implementation of a whole processing chain as described in Paragraph 4.3.1. Component that combines *fft_wrapper* and *nofft_slice* components.

- **arbiter:** System's arbiter. Component to generate as many Processing Chains as needed for the fully pipelined, efficient system operation, as described in Paragraph 4.3.1. Derives its 14-bit input from the system's ADC and feeds all Processing Chains with the appropriate data, as shown in Figure 4.8. One validity output bit, activated when the desired amount of accumulations has been executed, and a 32-bit output, which is the *big_adder* output, the spectrum point accumulated to be saved in Block RAM at each clock cycle.

4.3.2 LiDAR Direction Determination

The Coherent Detection Doppler LiDAR of the system is assumed to have two rotation motors, one controlling the azimuth and one the elevation angle of the telescope (Figure 4.2). Upon rotation, the motor's encoders are set to produce two 1-bit square-waves each, a combination of which can lead to both the turning direction and the rotation angle of the motor. The square-waves' frequency depends on the angular velocity of the rotation and may not surpass 15kHz.

Suppose a and b are the motor generated signals, for either the azimuth or the altitude motor. These signals, produced by the encoder during a stable velocity clockwise and counter-clockwise rotation are shown in Figure 4.10.

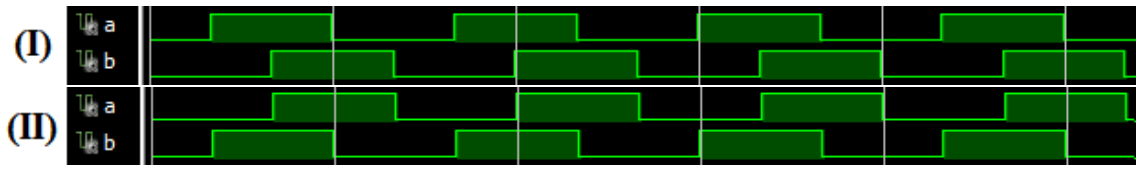


Figure 4.10: Motors' (I) clock- and (II) counter-clockwise rotation indicating signals

Their transitions should lead to the increase (when clockwise) or decrease (when counter-clockwise) of a counter value, which can afterwards be linearly transformed into degrees, depending on the motor's gears diameters ratio. These counter's value changes are presented in the truth Table 4.2. At the last column, "+" stands for counter increase by 1, "-" for counter decrease by 1, while "/" means no alteration to its value.

old A	A	old B	B	cnt
0	0	0	0	/
0	0	0	1	-
0	0	1	0	+
0	0	1	1	/
0	1	0	0	+
0	1	0	1	+
0	1	1	0	-
0	1	1	1	-
1	0	0	0	-
1	0	0	1	-
1	0	1	0	+
1	0	1	1	+
1	1	0	0	/
1	1	0	1	+
1	1	1	0	-
1	1	1	1	/

Table 4.2: Counter Value Change Truth Table

The VHDL code implements this counting process. The conversion from counter value to degrees is a simple multiplication by a constant and is executed in higher programming level (C language) on the ARM component of the SoC.

For further power consumption reduction, this part of the design can use lower clock frequency than the rest of the system, while input signals transitions occur at no more than 15kHz frequency and output results are not necessary, until the whole number of requested accumulations is executed to compute the accurate Doppler shift of the received signal.

4.4 Zynq Ultrascale 102+ MPSoC

The system's needs in memory resources and computation utilization, as well as the desire for trustworthy and direct communication with its environment and flexibility for the development of more circuitry, leads to the choice of a powerful, state-of-the-art multi-processor SoC.

Zynq Ultrascale 102+ (ZCU102) [33] is a Xilinx Multi-Processor System on Chip featuring a quad-core Arm® Cortex®-A53, a dual-core Cortex-R5F real-time processor and a Mali™-400 MP2 graphics processing unit based on Xilinx's 16nm FinFET+ programmable logic fabric. The ZCU102 supports all major peripherals and interfaces, enabling development for a wide range of applications.

It enables configuring via USB (onboard JTAG), dual Quad-SPI flash memory or SD

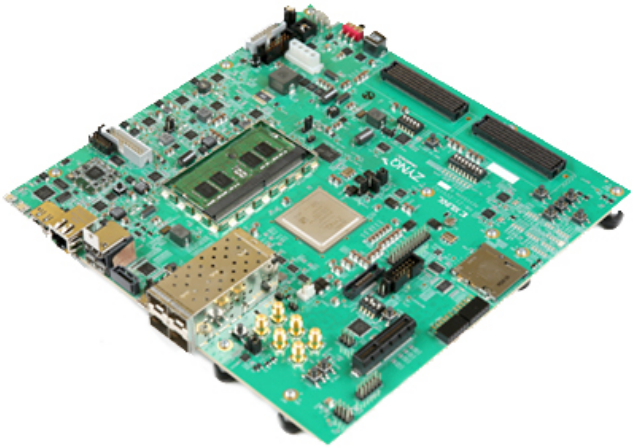


Figure 4.11: Zynq Ultrascale 102+ MPSoC [33]

card. The PS has a 4GB DDR4 64-bit SODIMM, while the PL a 512MB DDR4 component memory ([256 Mb x 16] devices) at 1200MHz / 2400Mbps DDR. IIC EEPROM of 8KB, a dual 64MB Quad SPI flash and an SD card slot are also available. Amongst others, it includes 5 Pl directional pushbuttons, 8 PL DIP switches, I2C bus switching, an IIC expansion connector, a UART To USB bridge, an RJ45 Ethernet connector, HDMI video input and output, while it also supports system and user clocks.

The MPSoc used is the Zynq UltraScale+ XCZU9EG-2FFVB1156 MPSoC, with the features of Table 4.3.

System Logic Cells (K)	600
Memory (Mb)	32.1
DSP Slices	2520
Maximum I/0 Pins	328

Table 4.3: ZCU102 features

Chapter 5

Experimental Results

5.1 Optimum Variables Exploration

A MATLAB exploration has been carried out for the determination of the optimum system variables described in Paragraph 4.2. The exploration is based on a series of simulations of the Wind LiDAR data acquisition and processing procedures explained in Chapter 4, inheriting code originally delivered in the "AIOLIKOS" Project [1] by Microlab ECE NTUA. Particularly, the variables examined are gate length L_g, FFT length N_{FFT} and the number of accumulations necessary to achieve specific accuracy standards. This number depends on the combination of the previous two and is from now on referred to as P.

The Mie and Rayleigh coefficients mentioned in Paragraph 2.4 were simulated to create the detected signal's expected Power Spectrum, after the addition of White Gaussian Noise. Even harsher noise conditions were also predicted and simulated by inserting a generic percentage of Intensity variation I_{var} and a generic absolute value of frequency variation f_{var} to the simulated backscattered pulses.

The modeling process includes an initial selection of the emitted radiation frequency F_{pul}. The backscattered radiation detected by the LiDAR's telescope is emulated in MATLAB. A pulse of L_g samples is created, with a random absolute variation of frequency around F_{pul}, depending on the value of f_{var}. The pulses' intensity also varies randomly, within a percentage dependent on I_{var} of the emitted pulse's intensity. This signal is randomly placed in some position of a N_{FFT} samples long vector of zeros. This simulates the zero-padding needed for the signal to be processed by the digital system's FFT component. Prior to its spectral analysis, the N_{FFT} samples vector is infected with White Gausian Noise of SNR determined by the $AWGN$ simulation variable, using MATLAB's *AWNG()* function to select noise levels in dB. The signal is now ready to be transformed into the frequency domain using the FFT. The main coefficient (Mie) seen in Figure 2.5 is the one obvious at the Power Spectrum after this MATLAB simulation. Rayleigh scattering coefficient is added afterwards to the Spectrum, in the form of a Gaussian distribution with the same mean and and higher standard deviation.

The simulation and analysis of a single noisy input of L_g samples leads to a totally unreliable determination of the signal's main frequency (Figure 4.4). The same routine is repeated P times, each one of which creating a new, randomly-varied from the initial emitted signal waveform, and then their output Spectra are accumulated.

The accumulation leads to the detection of a frequency f_d. This is achieved through the determination of the created Power Spectrum's maximum Density. It is, though, not precise to assume that this is necessarily the signal's frequency. If that was true, then the system designed would only be capable of detecting N_{FFT} discrete frequencies out of a continuous spectre of infinite possible F_{pul} values. A high accuracy approach is achieved using the technique of curve fitting (Paragraph 4.1). The data points approached at this case are the spectral density - frequency values corresponding to the maximum point of the Spectrum, as well as the points of the previous and next discrete frequencies. MATLAB tests carried out included up to 5^{th} grade polynomials. For the sake of simplicity, the usage of a 2^{nd} grade (Parabola) was decided, after observation of no significant enough improvement by using higher grades.

This method is used to increase the system discrete codomain's detection frequencies. That given, when an adequate amount of accumulations P is executed, the detected backscattered signal's frequency f_d approaches F_{pul} with about ten times higher accuracy.

The variable exploration was conducted through MATLAB simulations, altering the variables to be optimized (L_g, P), as well as the ones that affect them ($AWGN$, f_{var}, I_{var}). Required is an accuracy of at least 0.1 MHz in the frequency detection process, resulting to higher than 0.1 m/s accuracy in detecting the wind's velocity.

FFT Length was set to $N_{FFT} = 1024$ samples. A choice of a power of two is required to fully exploit Fast Fourier Transform algorithm's [17] structure. Early MATLAB tests resulted in unclear results for less N_{FFT} samples (256, 512), even for zero noise levels. F_{pul} estimation was much more frequently successful for $N_{FFT} \geqslant 1024$. This led to the 1024 samples choice for time and resources savings, at the expense of the higher resolution that more samples would be able to offer.

The simulations explore two main subjects:

I The relationship between gate length L_g and computation accuracy.

II The minimum accumulations P required to achieve the requested accuracy, for various noise conditions.

Absolute computation error is calculated repeating the procedure described rep consecutive times, and finding the distance between the emitted signal's frequency F_{pul} and the mean average detected frequency $\overline{f_d}$ (Equations 5.1.1 and 4.1.2).

$$\overline{f_{error}} = |F_{pul} - \overline{f_d}| \tag{5.1.1}$$

, where

$$\overline{f_d} = \frac{\sum_{i=1}^{rep}(f_d)_i}{rep} \tag{5.1.2}$$

5.1.1 Gate length effect on computation accuracy

These simulations examined the effect of different gate lengths L_g on the absolute computation error, for various noise conditions and a fixed number of accumulations P. Early simulations took place for a low number of experimental repetitions rep. This way, two choices of accumulation number P with different results were made (512 and 2048), for which the experiments were executed once more, this time for more repetitions $rep = 1000$, resulting to Figures 5.1 and 5.2.

(I) P = 512 accumulations

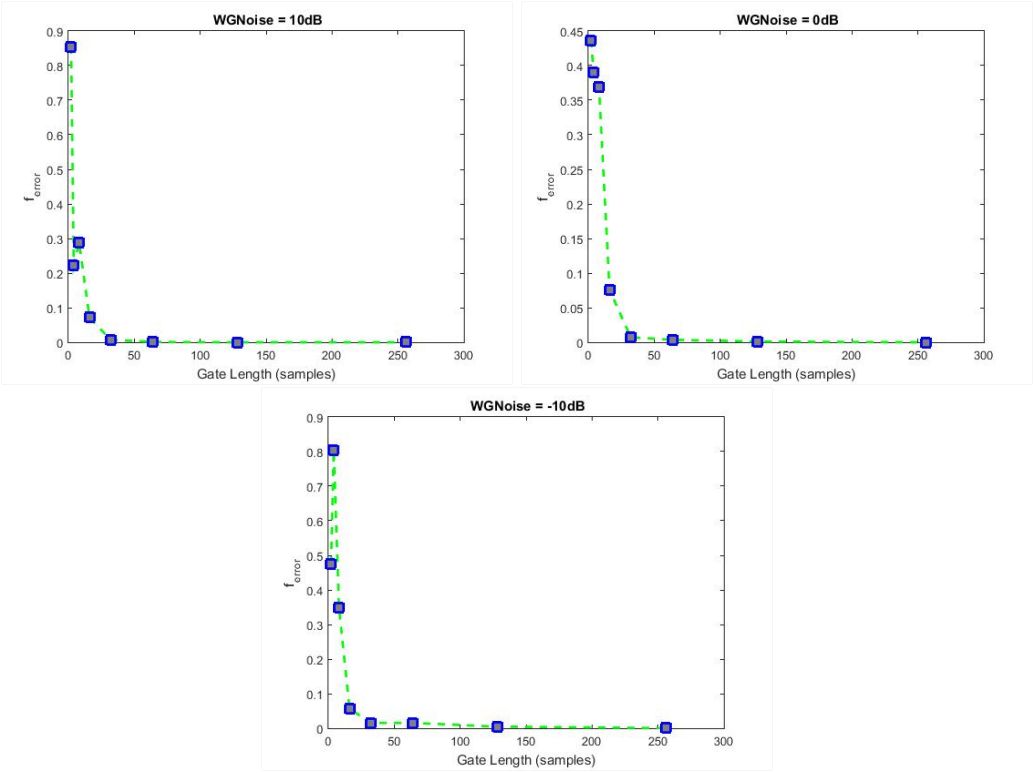


Figure 5.1: $f_{error} = f(L_g)$ for P = 512 accumulations

(II) P = 2048 accumulations

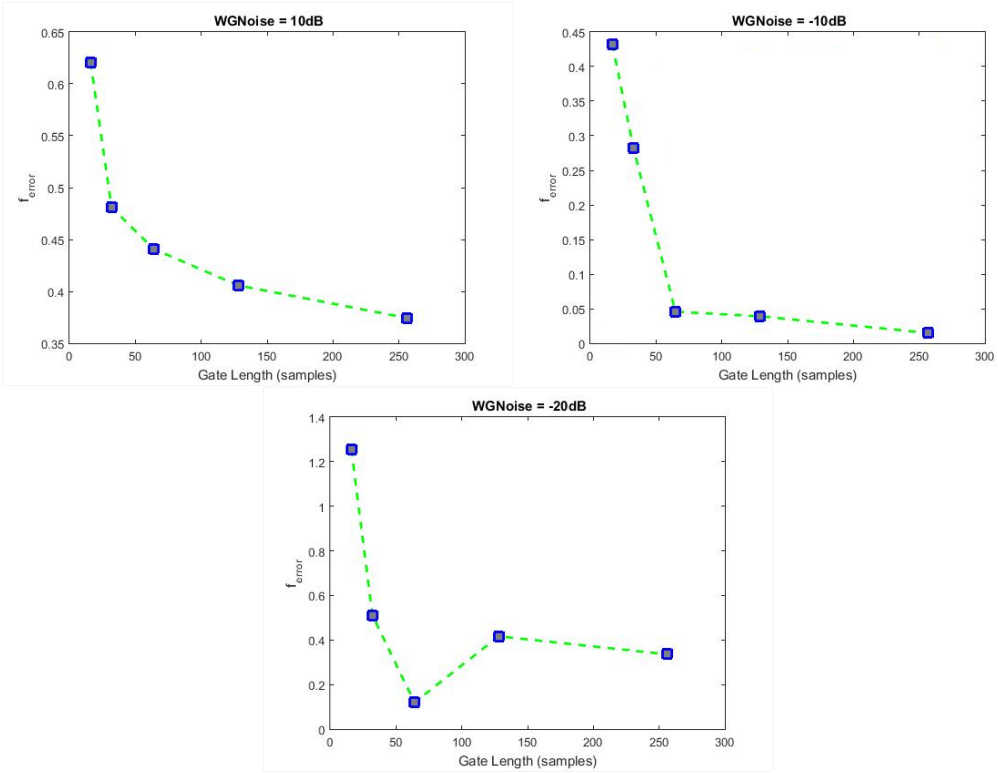


Figure 5.2: $f_{error} = f(L_g)$ for P = 2048 accumulations

The Figures 5.1 and 5.2, firstly, verify the dependence of the results' quality on the LiDAR's gate length L_g, regardless of the noise. Some spikes do exist; however, they only refer to small f_{error} differences due to the randomness inserted by the variables I_{var},f_{var} and the restricted number of experimental repetitions *rep*.

Deriving the optimum gate length L_g from those Figures is not that easy. The gate length chosen is the minimum L_g after which the chart's gradient noticeably decreases. In some charts of Figures 5.1 and 5.2 this happens at $L_g = 32$ and in some others at $L_g = 64$. These are, therefore, going to be the gate lengths chosen for further research and the implementation of the system.

5.1.2 Computing error in relation to the number of accumulations, for different noise conditions

Based on the results of the first simulations of Paragraph 5.1.1, further research includes fixed gate lengths L_g of 32 and 64 samples. For either case it is crucial to indicate the number of accumulations needed to achieve the requested 0.1MHz accuracy, depending of course on the noise levels. In this direction, the detection error was computed according to the Formula 5.1.1 for *rep*=1000 repetitions of the simulation. The following Figures include experimental results for the least favourable simulated noise conditions, while these conditions demand higher accumulation numbers P and are therefore more crucial. Stage

(I) includes simulation results for White Gaussian Levels of -10dB and -20dB, for couples of $[f_{var}, I_{var}] = [0.2 \text{ MHz}, 20\%]$ and [1 MHz , 50%]. Stage (II) inspects noise effect using more f_{var} values, which is the percentage of random variation of the simulated backscattered pulses, in regard to the emitted pulses' frequency F_{pul}, for a 3-D representation of the results.

(I) Computation of the necessary number of accumulations P

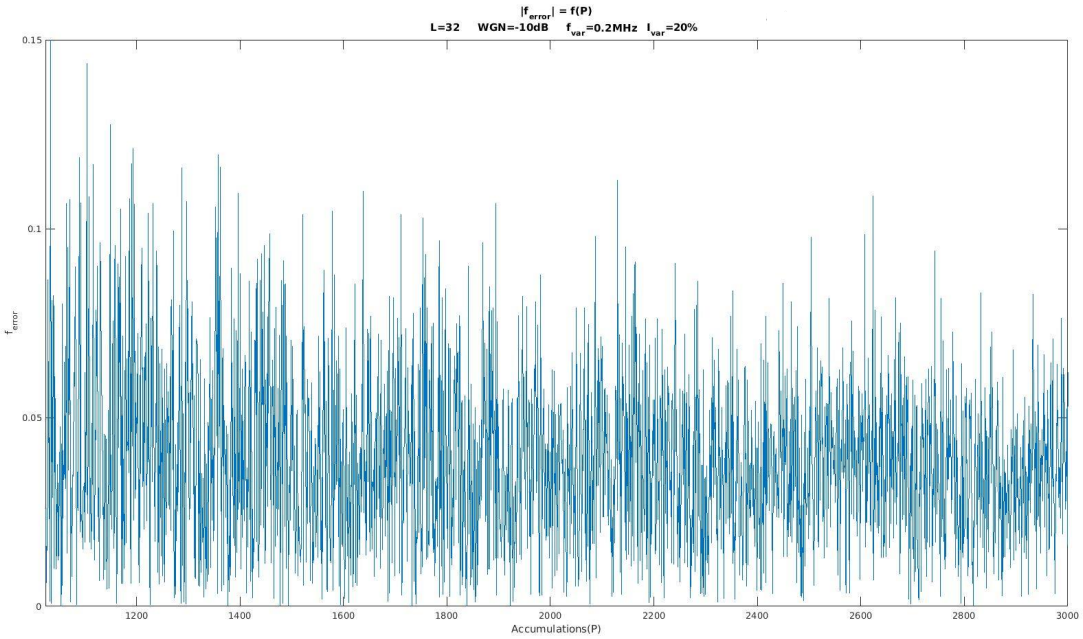


Figure 5.3: $f_{error} = f(P)$ with $L_g = 32$

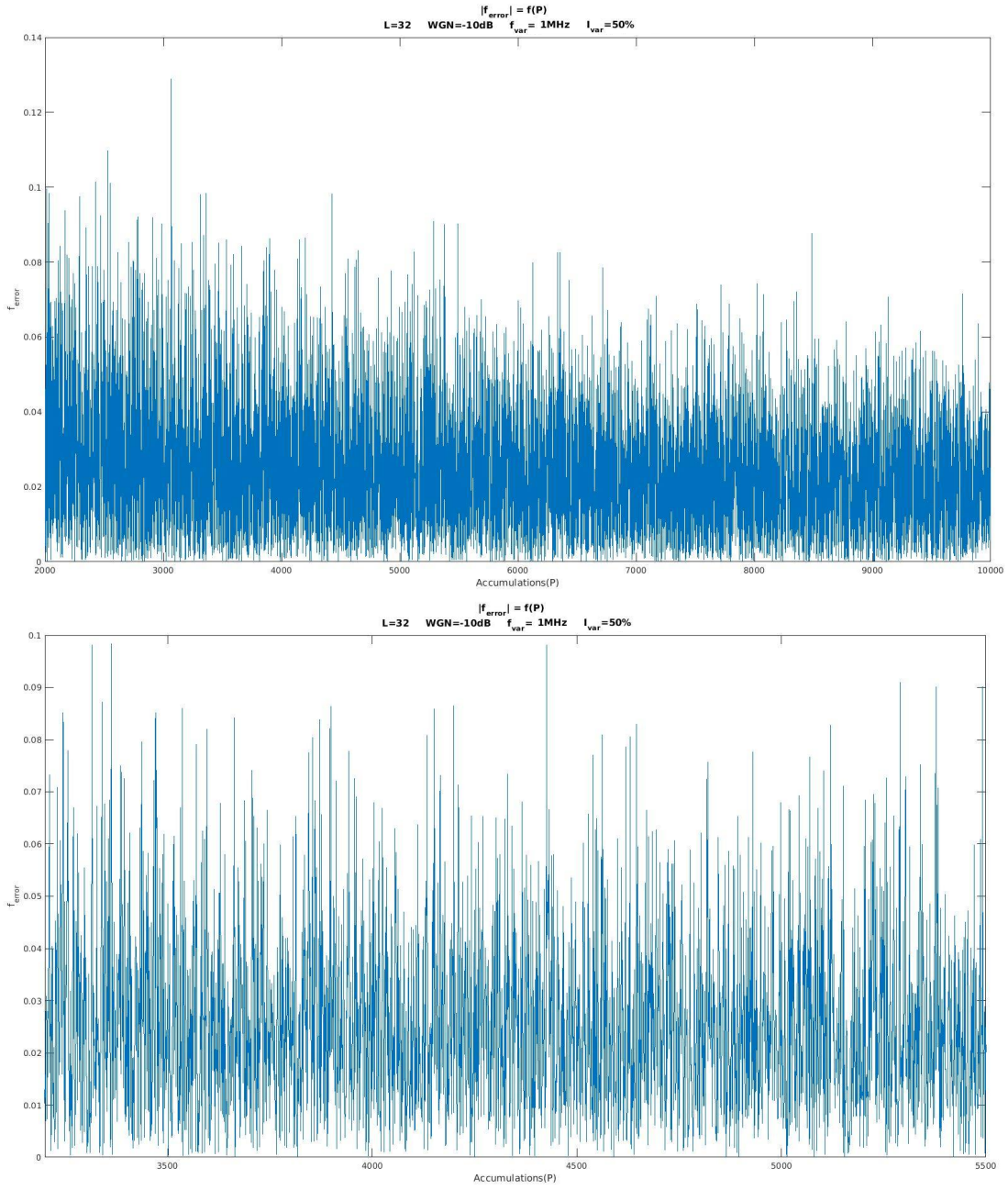


Figure 5.4: $f_{error} = f(P)$ with $L_g = 32$

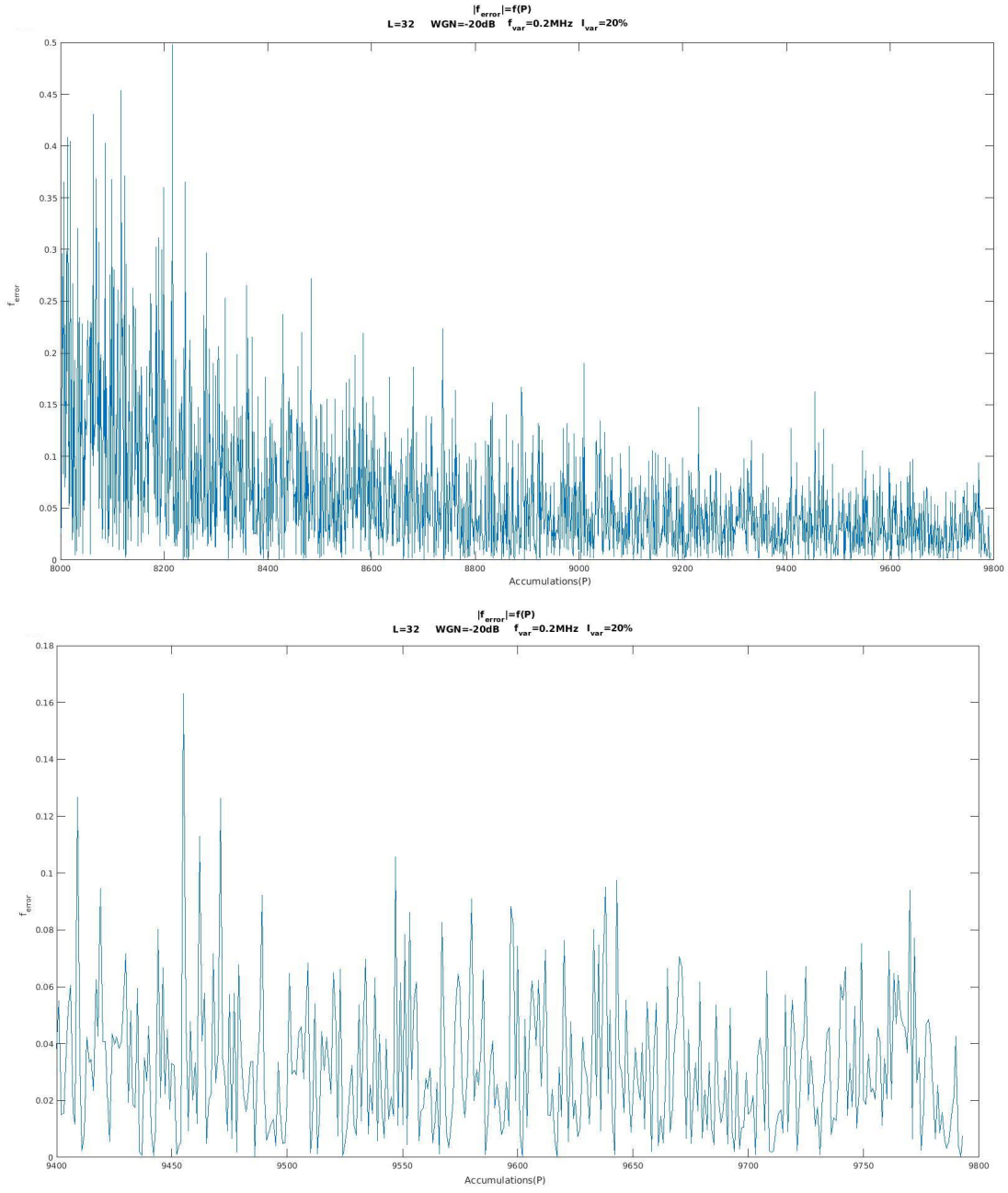


Figure 5.5: $f_{error} = f(P)$ with $L_g = 32$

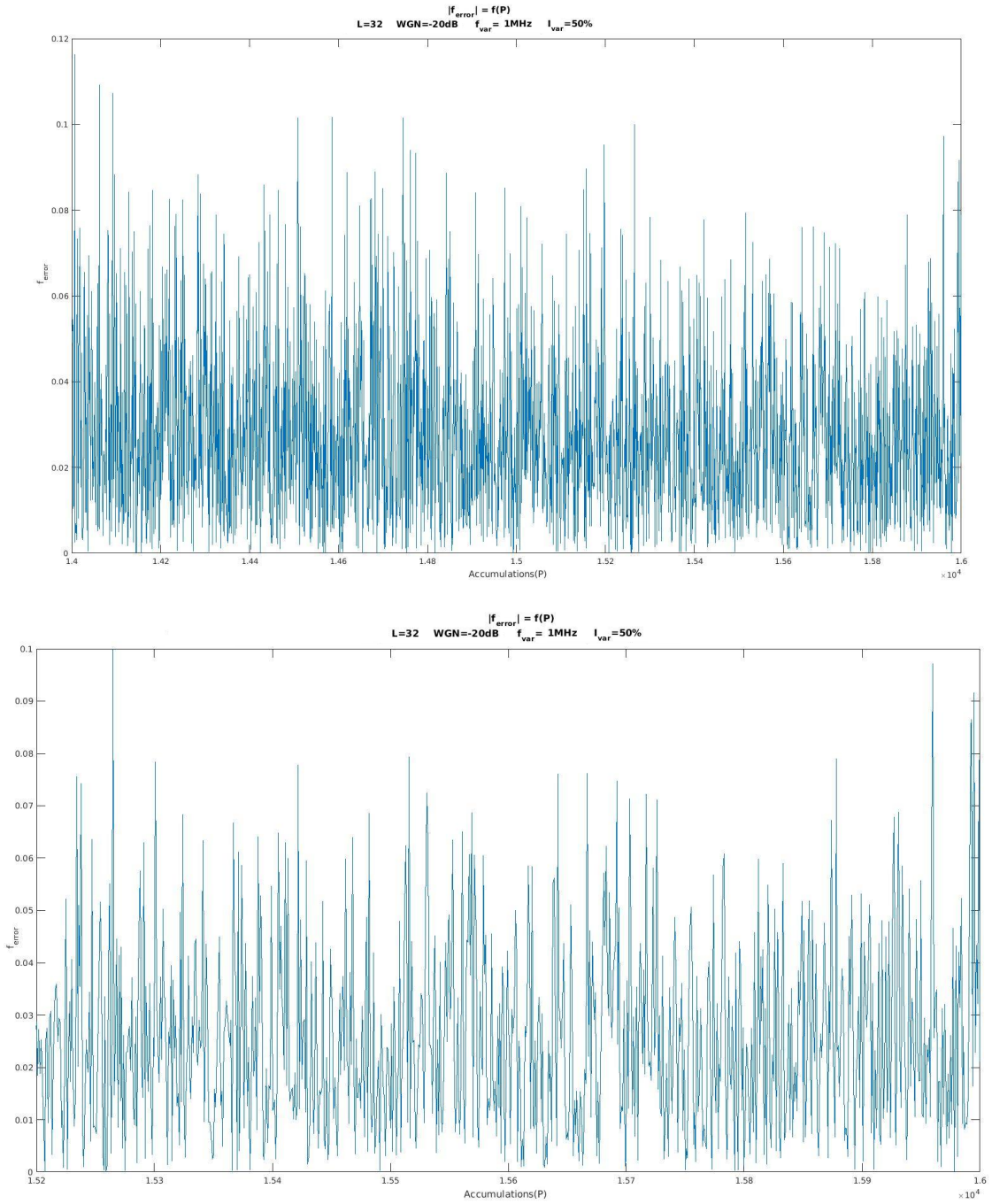


Figure 5.6: $f_{error} = f(P)$ with $L_g = 32$

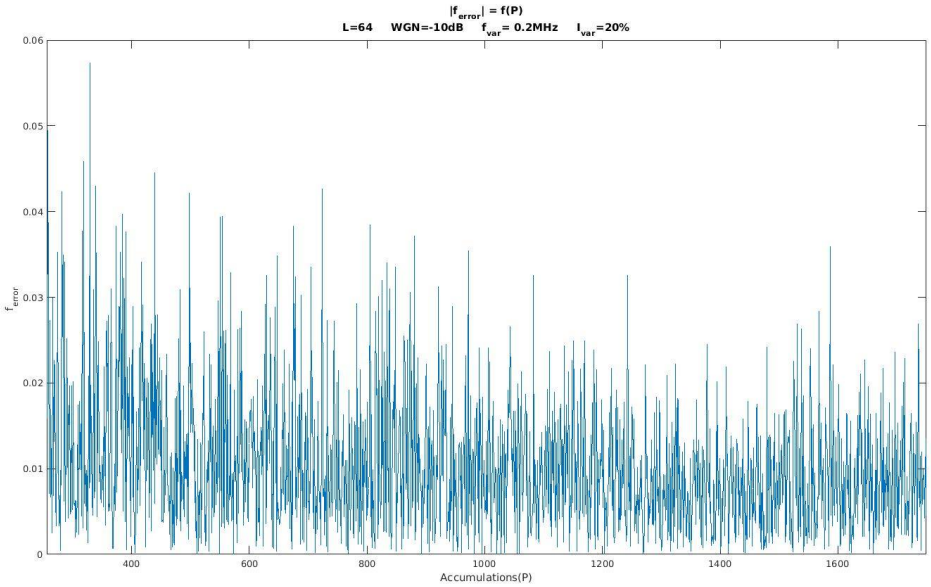


Figure 5.7: $f_{error} = f(P)$ with $L_g = 64$

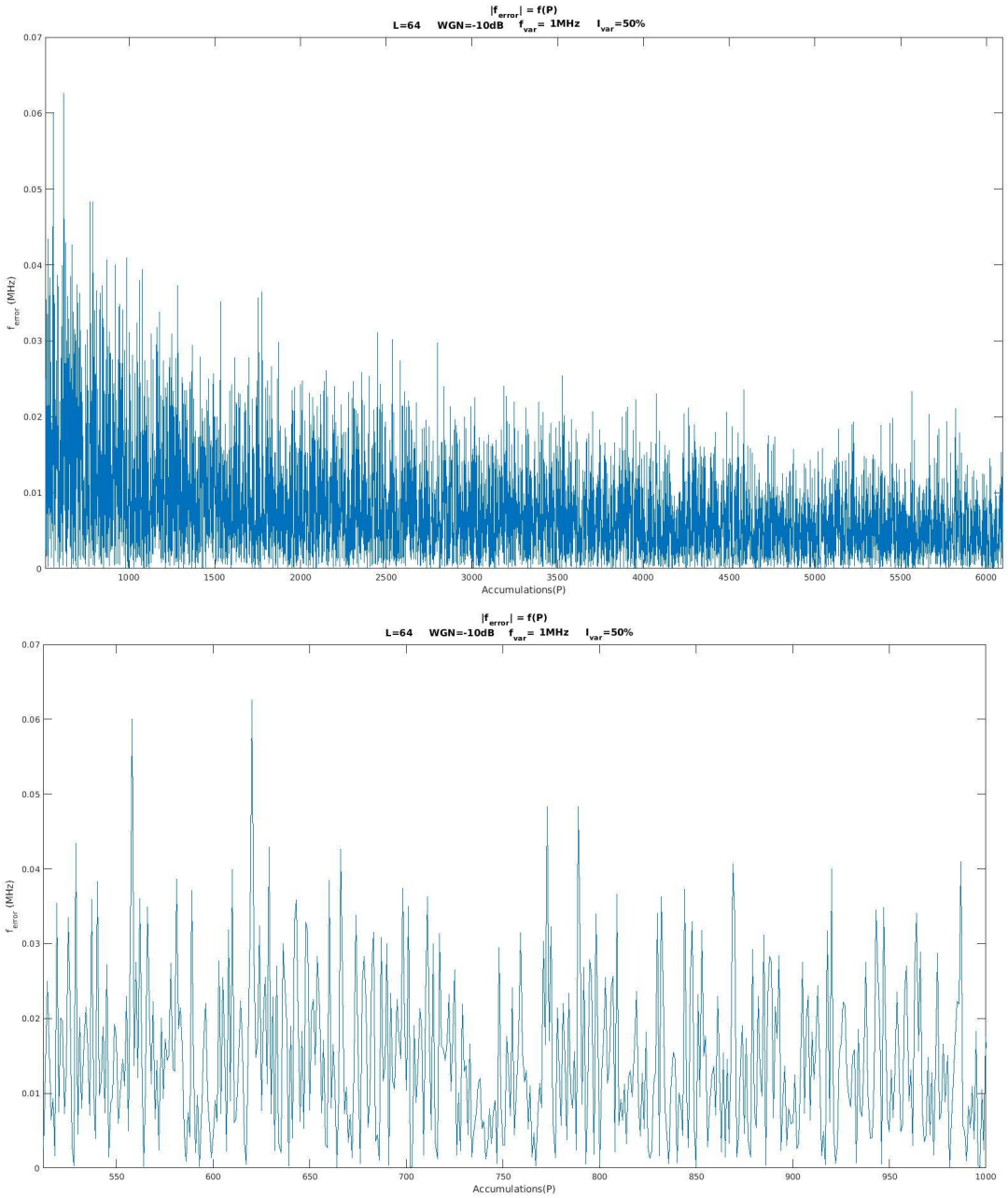


Figure 5.8: $f_{error} = f(P)$ with $L_g = 64$

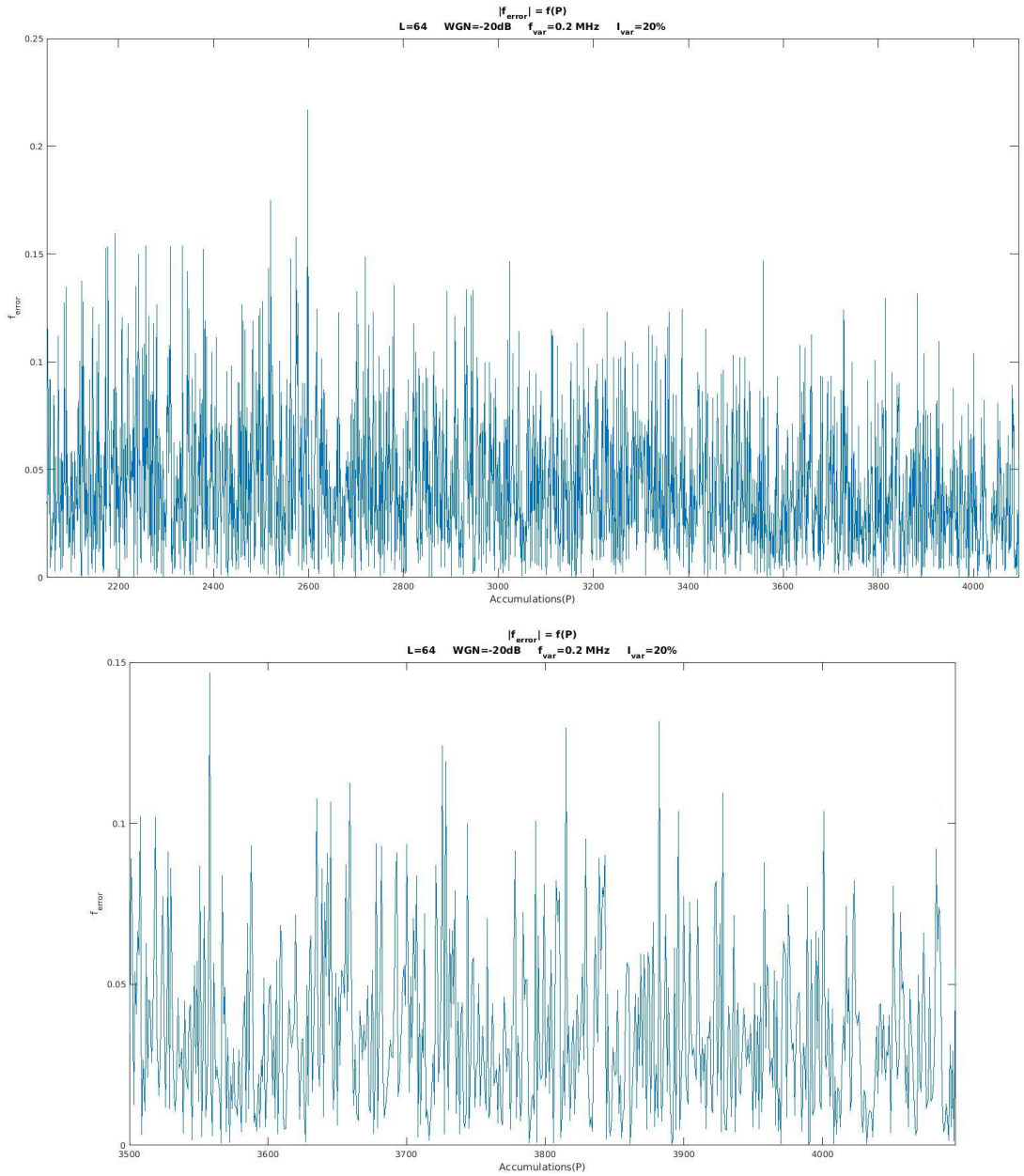


Figure 5.9: $f_{error} = f(P)$ with $L_g = 64$

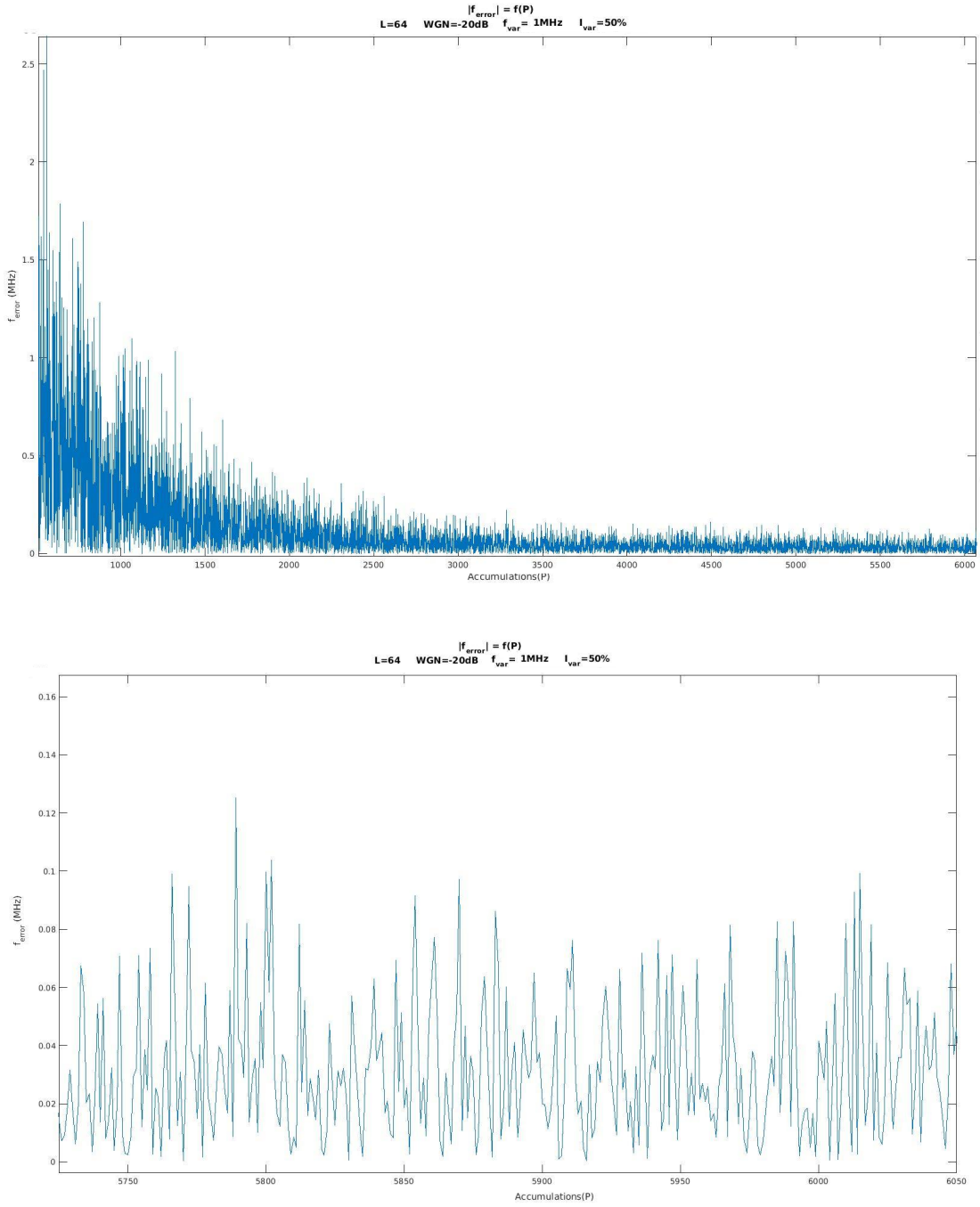


Figure 5.10: $f_{error} = f(P)$ with $L_g = 64$

As expected and explained in Chapter 3.1, a lower gate length L_g leads to harder to cope with input signal. It is, therefore, obvious that for the same noise conditions, the $L_g = 64$ system can reach the 0.1 MHz requested accuracy with a much lower accumulation rate than the $L_g = 32$ system. The simulations also confirm that regardless of the randomness of the noise, the absolute error of the detected frequency is qualitatively inversely proportional to the number of accumulations, up to a certain error point which

AWGN	**-10dB**		**-20dB**	
(I_{var}, f_{var})	**(0.2,0.2)**	**(0.5,1)**	**(0.2,0.2)**	**(0.5,1)**
$L_g = 32$	2630 accs	3010 accs	9550 accs	15260 accs
$L_g = 64$	330 accs	620 accs	3900 accs	6020 accs

Table 5.1: Accumulations P required to achieve 0.1MHz accuracy

cannot be further lowered.

The results of the previous simulations are gathered in Table 5.1, which holds the number of accumulations P needed to ensure the requested accuracy, depending on the simulated noise levels.

(II) Absolute computation error according to noise and number of spectral accumulations P

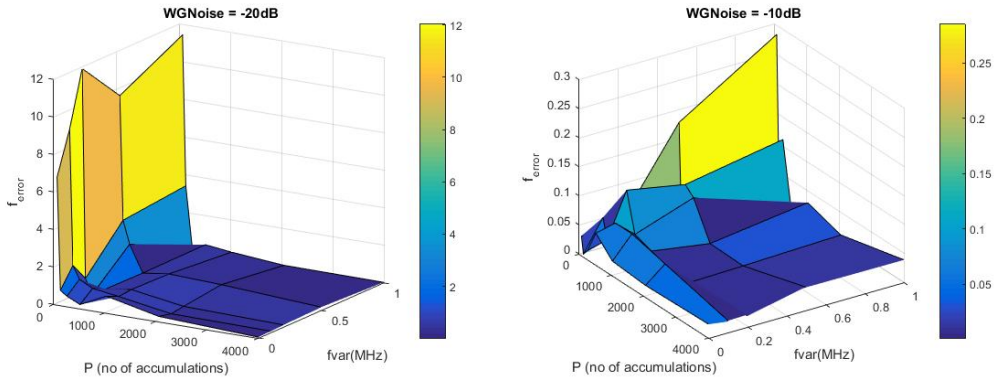


Figure 5.11: $f_{error} = f(P, f_{var})$ with $L_g = 32$

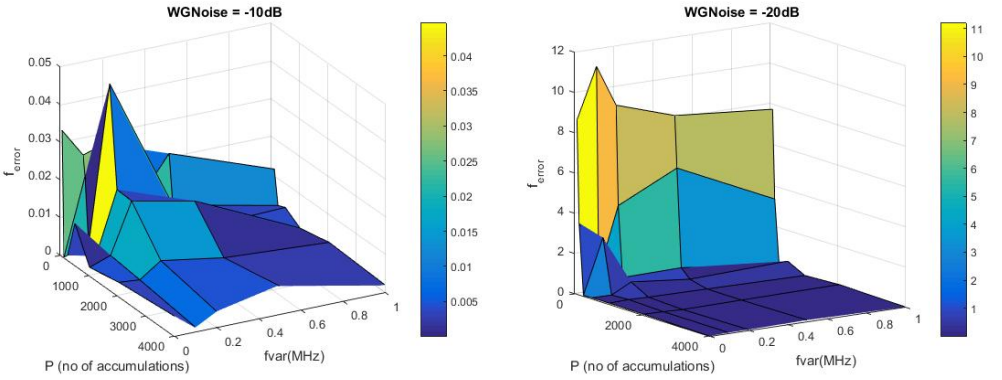


Figure 5.12: $f_{error} = f(P, f_{var})$ with $L_g = 64$

The Figures 5.11 and 5.12 depict the results of Experiment (II), as a 3-D extension of Experiment (I) for continuous noise level increase.

5.2 VHDL Design Verification

Using the gate lengths emerged from the MATLAB exploration of Paragraph 5.1, two VHDL designs were developed. The first uses L_g=32 samples and hence 15260 accumulations to reduce noise and render the signal's frequency detection possible even for unfavourable detection conditions. The second uses L_g=64 samples and 6020 accumulations.

5.2.1 $L_g = 32$ Design

This system is designed to support the processing of pulses of $L_g = 32$ samples each and the storage of their power spectra of $N_{FFT} = 1024$ samples each. The Equations 4.2.2 and 4.3.1 can be used to define the rest of the System's variable values.

Specifically, Equation 4.3.1 results to the generation of C = 32 Processing Chains to meet the pipelined operation requirement. Combining the other two equations, considering

- Wind LiDAR detection radius $R_D < 10km$
- ADC period $T_{ADC} = 2ns$
- speed of light $c = 3 * 10^8 \frac{m}{s}$

and supposing the pulse is ideal (zero width) arises the need of storing spectra for about 512 gates. The distribution of these spectra to 32 processing chains results to total of 16 Spectra to be stored in each processing chain. Each spectrum consists of 1024 32-bit samples, entirely fitting a whole Zynq Ultrascale 102+ RAM Block. Based on that, each Processing Chain also binds 16 RAM Blocks for the storage of its 16 Spectra. The last generic variable to be set is the number of accumulations needed, which is chosen to be $accs = 15260$ based on Table's 5.1 results. All these values are gathered in Tables 5.3 and 5.2.

The System's output waveforms after simulation on Vivado [32] are given in Figures 5.13, 5.14, 5.15

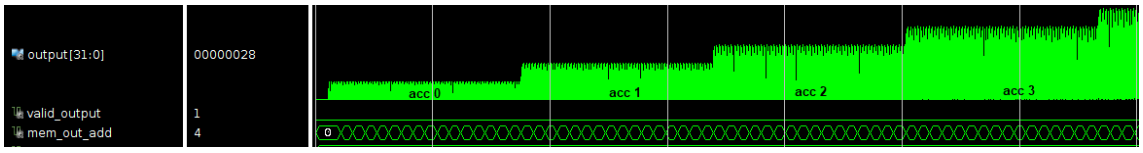


Figure 5.13: Spectral Accumulation L_g=32

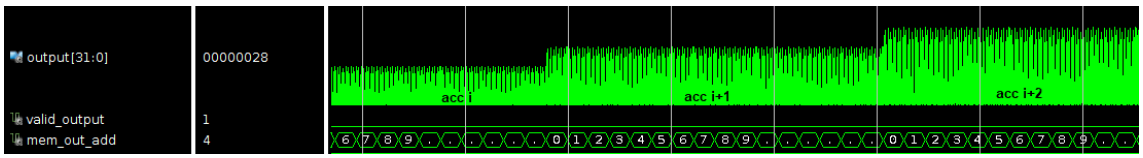


Figure 5.14: Spectral Accumulation L_g=32 (close up)

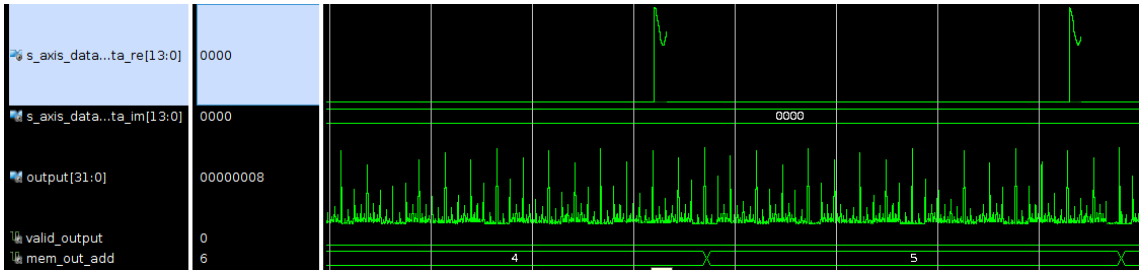


Figure 5.15: Single Gate Spectrum L_g=32

All simulation waveforms concern a single processing chain. Respective results can be found at each of the rest of the design's chains.

Figures 5.13 and 5.14 depict the procedure of spectral accumulation on PL. *Output* variable is the system's 32-bit output, which is the spectral point computed at each clock cycle. *Mem_out_add* is the variable that indicates the chain's gate corresponding to each spectrum, as explained in Paragraph 4.2 and Figure 4.9. *Valid_output* is a flag turned TRUE when the specified number of accumulations has been executed and the results are considered to be valid.

Figure 5.15 is a close up to a single Spectrum created. *S_axis_data_tdata_re* is the real part of the 14-bit input signal given to the FFT IP of this processing chain. It is the result of the zero padding technique explained in 4.2 and presented in Figure 4.8. The FFT produces outputs in a reversed digit order and the spectrum of Figure 5.15 is a result of just a few accumulations, which explains its unclear and sparse peaks. When the required amount of accumulations is executed, all spectra are given to the PS, where a C code reorders them to create a spectrum like the last of Figure 4.5. The wind velocity is, finally, computed using the parabola fitting technique.

5.2.2 $L_g = 64$ Design

This system is designed to support the processing of pulses of $L_g = 64$ samples each and the storage of their power spectra of $N_{FFT} = 1024$ samples each. Using the same equations and considerations as before, the $L_g = 64$ Design's constants are computed.

- $N_{FFT} = 1024$ samples
- $L_g = 64$ samples
- $C = 16$ processing chains
- $N_p = 256$ gates
- Block Spectra $= 16 \frac{Spectra}{Processing_Chain}$
- $Memory_Usage = 16 \frac{RAM_Blocks}{Processing_Chain}$

The System's output waveforms after simulation are given in Figures 5.17 and 5.16.

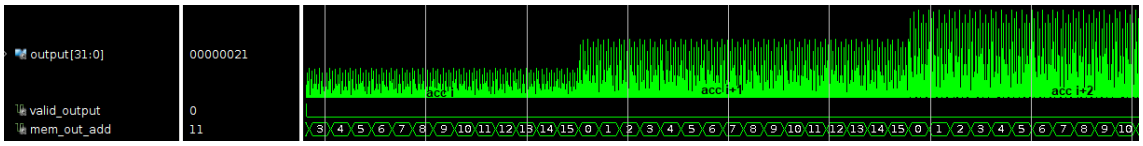


Figure 5.16: Spectral Accumulation L_g=64

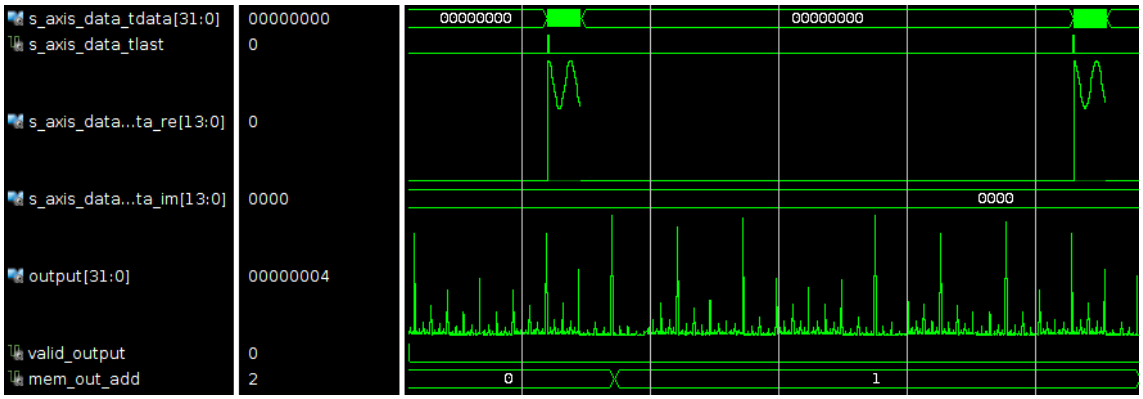


Figure 5.17: Single Gate Spectrum L_g=64

The signals of figures 5.16 and 5.17 are equivalent to the ones of Figures 5.14 and 5.15. The difference of gate length L_g results to more LiDAR received data being given as each processing chain's input.

5.2.3 Utilization and Timing

The resources percentage used among the resources available on Zynq Ultrascale 102+ SoC for each design and the systems' timing performance are presented in Tables 5.2 and 5.3. Execution Time is computed as:

$$ExecutionTime = ProcessingTimePerAccumulation * RequiredAccumulations \tag{5.2.1}$$

Required Accumulations (Equation 5.2.1) are derived from the last column of Table 5.1. Consecutive pulses are considered to be received without waiting time between each other. The percentages of Table 5.2 concern ZCU102+, among its available 274080 LUTs, 144000 LUTRAMs, 548160 FFs, 912 BRAMs and 2520 DSPs.

	CONSTANTS				**RESOURCES**				
L_g	N_{FFT}	C	N_P	$\frac{Spectra}{Chain}$	LUT	LUTRAM	DFF	DSP	BLOCKRAM
32	1024	32	512	16	32%	19%	32%	51%	63%
64	1024	16	256	16	16%	10%	16%	25%	32%

Table 5.2: Resources Utilization

	CONSTANTS				TIMING PERFORMANCE		
L_g	N_{FFT}	C	N_P	$\frac{Spectra}{Chain}$	Clock Frequency (MHz)	Processing Time per accumulation (μs)	Execution Time Max Accumulations Zero Wait (ms)
32	1024	32	512	16	500	35.08	535.32
64	1024	16	256	16	500	35.08	211.18

Table 5.3: Timing Performance

The Direction Detection component (Paragraph 4.3.2) has been tested initially on Zybo [18] and later on Zynq Ultrascale 102+ platform [33] (Paragraph 4.4), with the utilization for the latter shown in Table 5.4. DSPs and RAM Blocks were not used for this component.

Site Type	**Used**	**Available**	**Util%**
CLB LUTs	2923	274080	1.07
as Logic	2720	274080	0.99
as Memory	203	144000	0.14
CLB Registers	3210	548160	0.59
as Flip Flop	3108	548160	0.57
as Latch	102	548160	0.02

Table 5.4: Angle Detection Implementation Utilization

Chapter 6

Conclusions and Future Work

In conclusion, the thesis attempted to propose a system for Coherent Detection Doppler LiDAR digital processing and control and optimize the wind estimation averaging technique, through variable exploration. It presented an estimation of the requested number of accumulations, along with the equivalent execution time needed for the determination of wind velocity of several points in the radial direction of the LiDAR. It has also proposed two SoC FPGA digital designs that differ in execution time and resources utilization, further offering a different granularity in spatial accuracy. In cases when time is of the essence, the fastest of both designs may form the basis for the FPGA computation system, while applications requiring higher spatial accuracy may use the other design, at the expense of execution time increase. The generic nature of both the systems' variables also enables the usage of fewer accumulations to improve result delivery if favourable noise conditions allow this in terms of accuracy.

Undoubtedly, there is room for improvement. The designs are only tested under experimental conditions and have not yet processed real LiDAR data. System integration with the ADC, the PS, the PL and tuning its parameters are yet to take place. Real received signal processing and consequent testing of the implemented designs on the field with Coherent Doppler LiDAR are steps that have to be taken. Those tests will also enable the comparison with MATLAB simulation results.

The other parts of the proposed design are yet to be implemented, too. The IIC communication for LiDAR rotation and scanning control should be tested with the rotation motors and a GUI application, along with its connection to the SoC via Ethernet, must be engineered.

Future work also includes the implementation of different digital domain transformation algorithms and their integration into the system. Articles [27] propose the usage of means of spectral derivation alternative to the common Fast Fourier Transform, experimenting on theoretical and software basis. Hardware implemented systems of these may be of great interest, resulting to further acceleration, fewer resources and possibly to accuracy improvement.

Finally, the thesis system assumes the same ADC and FPGA operating frequencies.

Nevertheless, the design may also be extended to different sampling and computing rates, by creating a system of sample data storage and channeling, which will be able to support this difference. This, along with finding and improving the design's operational frequency limits, are two interesting and crucial matters to be further researched.

References

[1] Aiolikos. http://www.cres.gr/cres/pages/projects/projects_GR/espa_12.html.

[2] Azimuth. https://en.wikipedia.org/wiki/Azimuth#/media/File:Azimuth-Altitude_schematic.svg.

[3] Customer benefits by using fpgas. https://www.intel.com/content/www/us/en/products/programmable/fpga/new-to-fpgas/resource-center/overview.html.

[4] Fpga fundamentals. https://www.ni.com/en-us/innovations/white-papers/08/fpga-fundamentals.html.

[5] How to convert the analog signal to digital signal by adc converter. https://www.elprocus.com/analog-to-digital-adc-converter/.

[6] Soc fpgas. http://mycourses.ntua.gr/courses/ECE1207/document/soc-fpgas.pdf.

[7] What is a soc fpga. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ab/ab1_soc_fpga.pdf.

[8] T. Ando, E. Haraguchi, H. Ono, and A. Fijie. Mobile coherent doppler lidar prototype v 2 for wind sensing. 2018.

[9] C. Basics. Basics of the i2c communication protocol. https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/.

[10] Belmonte, A. Statistical model for fading return signals in coherent lidars. *Applied Optics*, 49(35):6737–6748, 2010.

[11] J. Beyon, T.-K. Ng, B. Lin, Y. Hu, and F. Harrison. High-speed on-board data processing for science instruments. page 90800L, 06 2014.

[12] A. Biasizzo and F. Novak. Hardware accelerated compression of lidar data using fpga devices. *Sensors (Basel, Switzerland)*, 13:6405–6422, 05 2013.

[13] Blackledge, J. M. *Digital Image Processing: Mathematical and Computational Methods*. Elsevier, Amsterdam, Netherlands, 2005.

[14] M. Chevrier. How to build a lidar system with a time-to-digital converter. 03 2017.

[15] X. Chu. Lidar remote sensing, lectures 2,3,20,21,24,25. `http://superlidar.colorado.edu/Classes/Lidar2016/`.

[16] A. Clifton, M. Boquet, E. Burin Des Roziers, A. Westerhellweg, M. Hofsass, T. Klaas, K. Vogstad, P. Clive, M. Harris, S. Wylie, E. Osler, B. Banta, A. Choukulkar, J. Lundquist, and M. Aitken. Remote sensing of complex flows by doppler wind lidar: Issues and preliminary recommendations. 12 2015.

[17] J. W. Cooley and J. W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Math. Comput.*, 19:297–301, 1965.

[18] Digilent. Zybo. `https://reference.digilentinc.com/reference/programmable-logic/zybo/start`.

[19] R. Frehlich and L. Cornman. Coherent doppler lidar signal spectrum with wind turbulence. *Applied optics*, 38:7456–66, 01 2000.

[20] N. Giordano. *College Physics: Reasoning and Relationships*. Cengage Learning, 2009.

[21] S. Kapp. Lidar-based reconstruction of wind fields and application for wind turbine control. 2017.

[22] Y. Lyu, L. Bai, and X. Huang. Chipnet: Real-time lidar processing for drivable region segmentation on an fpga, 08 2018.

[23] Papandrea, S., Jin, Y., Ristori, P., Otero, L., Nishizawa, T., Mizuno, A., ... Quel, E. Construction and first atmospheric observations of a high spectral resolution lidar system in argentina in the frame of a trinational japanese-argentinean-chilean collaboration. *Lidar Remote Sensing for Environmental Monitoring XV*, 2016.

[24] Salamitou, P., Dabas, A., Flamant, P. H. . Simulation in the time domain for heterodyne coherent laser radar. *Applied Optics*, 34(3):499–507, 1995.

[25] C. Slinger and M. Harris. Introduction to continuous-wave doppler lidar. 2012.

[26] Sun, W., Hu, Y., MacDonnell, D. G., Weimer, C., Baize, R. R. Technique to separate lidar signal and sunlight. *Optics Express*, 24(12):12949–12955, 2016.

[27] Wang, C., Xia, H., Liu, Y., Lin, S., Dou, X. Spatial resolution enhancement of coherent doppler wind lidar using joint time–frequency analysis. *Optics Communications*, 424:48–53, 2018.

[28] Wang, H., Barthelmie, R. J., Clifton, A., Pryor, S. C. Wind measurements from arc scans with doppler wind lidar. *Journal of Atmospheric and Oceanic Technology*, 32(11), 2015.

[29] Wu, D. L., Chae, J. H., Lambert, A., Zhang, F. F. Characteristics of caliop attenuated backscatter noise: implication for cloud/aerosol detection. *Atmospheric Chemistry and Physics*, 11(6):2641–2654, 2011.

[30] Xilinx. Cordic v6.0. `https://www.xilinx.com/support/documentation/ip_documentation/cordic/v6_0/pg105-cordic.pdf`.

[31] Xilinx. Fast fourier transform v9.0. `https://www.xilinx.com/support/documentation/ip_documentation/xfft/v9_0/pg109-xfft.pdf`.

[32] Xilinx. Vivado design suite. `https://www.xilinx.com/products/design-tools/vivado.html`.

[33] Xilinx. Zynq ultrascale+ mpsoc zcu102 evaluation kit. `https://www.xilinx.com/products/boards-and-kits/ek-u1-zcu102-g.html#hardware`.

[34] Zaher, A. *Recent Advances in Signal Processing*. IntechOpen, 2009.