Low Complexity, Parallel Algorithms, and Scalable Architectures for Real Time Coherent Optical OFDM Systems

Pramod Udupa

N° d'ordre : ANNÉE : 2014

THÈSE / UNIVERSITÉ DE RENNES 1

sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Traitement du Signal et Télécommunications

École doctorale : MATISSE

présentée par

Pramod UDUPA

préparée à l'unité de recherche : IRISA - UMR 6074

Institut de recherche en informatique et systèmes aléatoires - CAIRN

École Nationale Supérieure des Sciences Appliquées et de Technologie

---

Algorithmes parallèles et architectures évolutives de faible complexité pour systémes optiques OFDM cohérents temps réel

Low Complexity, Parallel Algorithms and Scalable Architectures for Real Time Coherent Optical OFDM Systems

Composition du jury :

Lilian BOSSUET
Maître de Conférences HDR, Télécom Saint-Etienne Université Jean Monnet / Examinateur

Erwan PINCEMIN
Ingénieur, Orange Labs
Examinateur

Michel JEZEQUEL
Professeur, Electronics Department Télécom Bretagne / Examinateur

Emmanuel BOUTILLON
Professeur, Lab-STICC
Université de Bretagne Sud / Rapporteur

Christophe JEGO
Professeur, IMS, Bordeaux
Universités - IPB/ENSEIRB-MATMECA / Rapporteur

Olivier SENTIEYS
Directeur de Recherche, IRISA/INRIA
Université de Rennes 1 / Directeur de thèse

Laurent BRAMERIE
Ingénieur de Recherche, FOTON, ENSSAT
Université de Rennes 1 / Co-directeur de thèse

To,
AMMA *and* ***APPA***

Acknowledgements

I wish to express profound gratitude to my thesis director Prof. Olivier Sentieys for guiding me throughout the time span of this work. I am grateful to him for his expert advice and time for thesis discussions and begin available for questions/clarifications at all times. The meetings which I had helped with many aspects of work. His suggestions and comments improved the quality of the thesis report.

I also wish to express my thanks to co-director Mr. Laurent Bramerie for his guidance in the topics related to Optical Communication Systems. His evaluation of ideas from optical systems point of view helped in validation of the work. I am very thankful for many discussions on optical experiments.

I am thankful to my colleague Rémi Pallas for bringing up the real-time FPGA development system and then integrating my architecture implementation on it. It was a very hard job and he was a very good experience working with him for all the three years. I also acknowledge Arnaud Carer for his help in setting up real-time FPGA platform and discussions regarding implementation.

Thanks are also due to faculty members, senior researchers and rest of my colleagues at ENSSAT with whom I mutually shared ideas and had discussions. I am thankful especially to Mme. Nathalie Caradec for her French classes. I would also like to acknowledge the assistance of administrative staff of ENSSAT due to which I had a pleasant work environment. Further I wish to recollect with fondness, the memorable association I developed with my friends Hai, Jérémy, Karthik, Nhan, Rémi, Rengarajan, Stéphane, Vaibhav, Vinh and Vivek.

This thesis has been made possible thanks to the funding by 100GFLEX project and facilities extended by IRISA/INRIA, including travel assistance to attend conferences, for which I remain grateful.

I wish to express my deep sense of gratitude to my parents for their encouragement in all phases of my academic and professional career that in the first instance enabled me take up doctoral studies. They sharing my goal of acquiring a doctorate only added to my inspiration to complete the doctoral program successfully. I am also grateful to the rest of my family members and friends whose constant support and words of encouragement enabled me to focus on my work.

Last, but not the least, I thank the members of the jury for agreeing to make a critical assessment of the dissertation and suggesting improvements to the thesis that enhanced its quality.

19th June 2014

Pramod UDUPA
Lannion, France

Résumé

Les systèmes de communications optiques à très haut débit sont construits à partir des techniques de pointe pour la détection, la modulation et la compensation de dispersion tels que, la détection cohérente, les modulations multi-porteuses orthogonales (OFDM) et la compensation électronique des dispersions (EDC). La réapparition de la détection cohérente dans les systèmes de communication optique a été rendue notamment possible par les progrès dans les circuits numériques dans les technologies avancées. La détection cohérente possède une meilleure sensibilité pour la détection du signal par rapport aux méthodes de détection directe. Elle permet d'utiliser des transmissions à double polarisation et conserve les informations de phase du signal optique et les transfert dans le domaine électrique. L'utilisation de la modulation OFDM fournit une flexibilité significative et l'utilisation efficace de la bande passante allouée. En raison de la disponibilité des informations de phase dans le domaine numérique, les processeurs DSP de faible coût peuvent être utilisés pour la compensation des dispersions dans le domaine numérique qui rend la solution flexible et reconfigurable. Mais, l'introduction du système CO-OFDM *(Coherent-Optical OFDM)* à la place de système de IM-DD *(Intensity Modulation-Direct Detection)* augmente significativement le coût du système avec un plus grand nombre de composants optiques et une quantité plus élevée de ressources électroniques requises pour la réception du signal. À l'heure actuelle, cela rend cette solution uniquement justifiable pour des transmissions à longue portée, même si le nombre de ressources par rapport à un système mono-porteuse à détection cohérente et modulation à quatre états (DP-CO-QPSK). Le choix de l'algorithme et l'optimisation de la précision des calculs en virgule fixe de l'architecture peuvent réduire de façon significative les ressources nécessaires pour la réalisation de systèmes CO-OFDM.

Dans cette thèse, des algorithmes à faible complexité et des architectures parallèles et efficaces sont explorés pour les systèmes CO-OFDM. Tout d'abord, des algorithmes de faible complexité pour la synchronisation et l'estimation du décalage en fréquence en présence d'un canal dispersif sont étudiés. Un nouvel algorithme de synchronisation temporelle à faible complexité qui peut résister à grande quantité de retard dispersif est proposé et comparé par rapport aux propositions antérieures. Ensuite, le problème de la réalisation d'une architecture parallèle à faible coût est étudié et une architecture parallèle générique et évolutive qui peut être utilisée pour réaliser tout type d'algorithme d'auto-corrélation est proposé. Cette architecture est ensuite étendue pour gérer plusieurs échantillons issus du convertisseur analogique/numérique (ADC) en parallèle et fournir une sortie qui suive la fréquence des ADC. L'évolutivité de l'architecture pour un nombre plus élevé de sorties en parallèle et les différents types d'algorithmes d'auto-corrélation sont explorés.

Une approche d'adéquation algorithme-architecture est ensuite appliquée à l'ensemble de la chaîne de l'émetteur-récepteur CO-OFDM. Du côté de l'émetteur, un algorithme IFFT à *radix*-2^2 est choisi pour et une architecture parallèle Multipath Delay Commutator (MDC) Feed-forward (FF) est choisie car elle consomme moins de ressources par rapport aux architectures MDC-FF en *radix*-2/4. Au niveau du récepteur, un algorithme efficace pour l'estimation du *Integer CFO* est adopté et implémenté de façon optimisée sans l'utilisation de multiplicateurs complexes. Une rÃĭduction de la complexité matérielle est obtenue grâce à la conception d'architectures efficaces pour la synchronisation temporelle, la FFT et l'estimation du CFO. Une exploration du compromis entre la précision des calculs en virgule fixe et la complexité du matériel est réalisée pour la chaîne complète de l'émetteur-récepteur, de façon à trouver des points de fonctionnement qui n'affectent pas le taux

d'erreur binaire (TEB) de manière significative. Les algorithmes proposés sont validés à l'aide d'une part d'expériences *off-line* en utilisant un générateur AWG *(arbitrary waveform generator)* à l'émetteur et un oscilloscope numérique à mémoire (DSO) en sortie de la détection cohérente au récepteur, et d'autre part un émetteur-récepteur temps-réel basé sur des plateformes FPGA et des convertisseurs numériques. Le TEB est utilisé pour montrer la validité du système intégré et en donner les performances.

Abstract

Coherent Optical-OFDM (CO-OFDM) communication system is built on most advanced techniques for detection, modulation and dispersion compensation *viz.*, coherent detection, orthogonal multi-carrier modulation (OFDM) and electronic dispersion compensation (EDC). The re-emergence of coherent detection in optical communication systems was made possible by the advancement in very high rate digital circuits. Coherent detection (CoD) has higher sensitivity for signal detection compared to direct detection (DD) methods. It enables use of dual-polarization transmission and it preserves phase information of optical signal and passes it to electrical domain. The use of OFDM modulation provides significant flexibility and efficient use of allocated bandwidth. Due to availability of phase information in digital domain, low cost digital signal processing (DSP) processors can be used for dispersion compensation in digital domain, which makes the solution flexible and re-configurable. But, the introduction of CO-OFDM system in place of older intensity modulation-direct detection (IM-DD) system significantly increases the cost of the system, i.e. higher number of optical components and higher amount of electronic resources are required for reception of the signal. Due to increase of resources both in optical and electronic domain, it is justifiable for only long-range transmission distances. The choice of algorithm, architecture and fixed-point optimization play a significant role in reduction of electronic resources required for realization of CO-OFDM systems.

In this thesis, low-complexity algorithms and architectures for CO-OFDM systems are explored. First, low-complexity algorithms for estimation of timing and carrier frequency offset (CFO) in dispersive channel are studied. A novel low-complexity timing synchronization algorithm, which can withstand large amount of dispersive delay, is proposed and compared with previous proposals. Then, the problem of realization of low-complexity parallel architecture is studied. A generalized scalable parallel architecture, which can be used to realize any auto-correlation algorithm, is proposed. It is then extended to handle multiple parallel samples from ADC and provide outputs, which can match the input ADC rate. The scalability of the architecture for higher number of parallel outputs and different kinds of auto-correlation algorithms is explored.

An algorithm-architecture approach is then applied to the entire CO-OFDM transceiver chain. At the transmitter side, radix-2^2 algorithm for IFFT is chosen and parallel Multipath Delay Commutator (MDC) Feed-forward (FF) architecture is designed which consumes lesser resources compared to MDC FF architectures of radix-2/4. At the receiver side, efficient algorithm for Integer CFO estimation is adopted and efficiently realized without the use of complex multipliers. Reduction in complexity is achieved due to efficient architectures for timing synchronization, FFT and Integer CFO estimation. Fixed-point analysis for the entire transceiver chain is done to find fixed-point sensitive blocks, which affect bit error rate (BER) significantly. The algorithms proposed are validated using optical experiments by the help of arbitrary waveform generator (AWG) at the transmitter and digital storage oscilloscope (DSO) and Matlab at the receiver. BER plots are used to show the validity of the system built. Hardware implementation of the proposed synchronization algorithm is validated using real-time FPGA platform.

Contents

List of Figures

List of Tables

List of Abbreviations

100 GbE	100 Gigabit Ethernet
ADC	Analog to Digital Converter
ASE	Amplified Spontaneous Noise
ASIC	Application Specific Integrated Circuit
AWG	Arbitrary Waveform Generator
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPF	Band Pass Filter
BPSK	Binary Phase Shift Keying
CD	Chromatic Dispersion
CE	Channel Estimation
CFO	Carrier Frequency Offset
CMZM	Complex Mach-Zender Modulator
CO	Coherent Optical
CO-DP-QPSK	Coherent Optical Dual Polarization QPSK
CO-OFDM	Coherent Optical-OFDM
CoD	Coherent Detection
CP	Cyclic Prefix
CPE	Common Phase Error
DAC	Digital to Analog Converter
DCF	Dispersion Compensated Fiber
DD	Direct Detection
DFT	Discrete Fourier Transform
DGD	Differential Group Delay
DP-QPSK	Dual Polarization QPSK

DPSK	Differential Phase Shift Keying
DSO	Digital Storage Oscilloscope
DSP	Digital Signal Processing
DWDM	Dense Wavelength Division Multiplexing
EB	Eta Bytes (10^{18})
ECL	External Cavity LASER
EDC	Electronic Dispersion Compensation
EDFA	Erbium-Doped Fiber Amplifier
ENOB	Effective Number of Bits
EVM	Error Vector Magnitude
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
FSF	Frequency Selective Fading
FSM	Finite State Machine
FTTH	Fiber-to-the-Home
GB	Giga Bytes (10^9)
GOPS	Giga Operations per second
GVD	Group Velocity Dispersion
HLS	High Level Synthesis
ICI	Inter Carrier Interference
IDFT	Inverse Discrete Fourier Transform
IEEE	Institute of Electrical and Electronic Engineers
IFFT	Inverse Fast Fourier Transform
IIR	Infinite Impulse Response
IM	Intensity Modulation
IM-DD	Intensity Modulation-Direct Detection
ISI	Inter Symbol Interference
ITU	International Telecommunication Union
LAN	Local Area Network

LASER	Light Amplification by Stimulated Emission of Radiation
LD	LASER Diode
LEAF	Large Effective Area Fiber
LMS	Least Mean Squares
LO	Local Oscillator
LPF	Low Pass Filter
LS	Least Squares
LTE	Long Term Evolution
MAN	Metropolitan Area Network
MB	Multi-Band
MB-OFDM	Multi-band OFDM
MDC	Multipath Delay Commutator
MDF	Multipath Delay Feedback
MHz	Mega Hertz
MIMO	Multi Input Multi Output
MLSE	Maximum Likelihood Sequence Estimation
MOPS	Million of Operations per second
MSE	Mean Square Error
MZM	Mach-Zender Modulator
N-WDM	Nyquist Wavelength Division Multiplexing
NGI CO-OFDM	No-Guard Interval CO-OFDM
NLMS	Normalized Least Mean Square
NRZ	Non Return-to-Zero
OBM-OFDM	Orthogonal Band Multiplexed OFDM
OBPF	Optical BPF
OEO	Optical-to-Electrical-to-Optical
OFDM	Orthogonal Frequency Division Multiplexing
OOK	On-Off Keying
OSA	Optical Spectrum Analyzer
OSNR	Optical Signal to Noise Ratio
PAPR	Peak to Average Power Ratio
PB	Peta Bytes (10^{15})
PBC	Polarization Beam Combiner

PBS	Polarization Beam Splitter
PCD	Polarization Dependent Chromatic Dispersion
PD	Photo Diode
PDM	Polarization Division Multiplexing
PDM-CO-OFDM	Polarization Division Multiplexing-Coherent Optical-Orthogonal Frequency Division Multiplexing
PDM-QPSK	Polarization Division Multiplexing-Quadrature Phase Shift Keying
PDR	Polarization Diverse Receiver
PMD	Polarization Mode Dispersion
PRBS	Pseudo Random Bit Sequence
PS	Polarization Scrambler
PSCF	Pure Silica Core Fiber
PSK	Phase Shift Keying
PSP	Principle State of Polarization
PSS	Primary Synchronization Signal
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RGI CO-OFDM	Reduced Guard Interval CO-OFDM
RMSE	Root Mean Square Error
ROADM	Reconfigurable Optical Add Drop Multiplexer
RS	Reed-Solomon
RZ	Return-to-Zero
RZ-DQPSK	Return-to-Zero Differential Quadratic Phase Shift Keying
SCO	Sampling Clock Offset
SDC	Single Path Delay Commutator
SDF	Single Path Delay Feedback
SNR	Signal to Noise Ratio
SOP	State of Polarization

SOPMD	Second Order PMD
SPM	Self Phase Modulation
SQNR	Signal Quantization to Noise Ratio
SSMF	Standard Single Mode Fiber
TB	Tera Bytes (10^{12})
TS	Training Symbol
ULH	Ultra Long Haul
VNI	Visual Networking Index
VOA	Variable Optical Attenuator
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
XPM	Cross Phase Modulation
ZF	Zero Forcing

Chapitre 0

Résumé étendu

0.1 Système de communications optiques OFDM à détection cohérente

L'arrivée massive des équipements connectés tels que les *smartphones*, tablettes, etc., et l'augmentation de la demande des services basés sur la vidéo, impliquent une augmentation exponentielle de la bande passante des réseaux, ce qui augmente la pression sur tous les nœuds du réseau Internet. Les réseaux de communication optique basés sur des fibres optiques SMF *(single-mode optical fiber)* sont utilisés pour transporter les données à haut débit sur de longues distances *(long-haul)*, des distances moyennes *(metro)* ou des distances plus courtes *(Fiber-to-the-Home (FTTH))*. Les réseaux optiques peuvent donc être divisés en trois partie majeures comme illustrées sur la Figure 1.

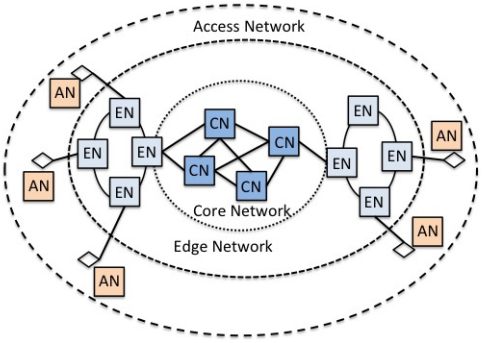

FIGURE 1: Architecture typique d'un réseau optique. *CN - Core Node, EN - Edge Node, AN - Access Node*

Pour transmettre des données sur une fibre optique simple mode (SMF), des techniques de modulation et détection directes IM-DD *(intensity modulation-direct detection)* sont utilisées pour obtenir des débits de 10 Gb/s sur des réseaux longues distances. Pour supporter l'augmentation de la demande en débit, l'objectif est de supporter des liaisons à 100 Gb/s [6]. Ceci ne peut être atteint avec le schéma IM-DD de façon efficace à cause de phénomènes tels que la dispersion chromatique CD *(chromatic dispersion)* ou la dispersion du mode de polarisation PMD *(polarization mode dispersion)* qui apparaissant à de telles

vitesses. Pour atteindre de tels débits, la détection cohérente CoD *(coherent detection)* a été introduite dans les systèmes de communications optiques et rendue possible grâce aux progrès des circuits intégrés èlectroniques. La détection cohérente [7][8][9] offre des avantages grâce à sa meilleure sensibilité de détection, à un débit symbole plus élevé, à l'utilisation de la double polarisation *(dual polarization)* et, de façon plus importante, à la conservation de l'information de phase et d'amplitude entre les domaines optiques et électroniques, ce qui ouvre la possibilité d'utiliser de puissants algorithmes de traitement numérique du signal pour la compensation électronique des dispersions EDC *(electronic dispersion compensation)* à un faible coût et de façon flexible grâce aux DSP. L'utilisation des modulations OFDM *(orthogonal frequency division multiplexing)* a été proposée pour être utilisée conjointement avec la CoD pour atteindre des débits de 100 Gb/s avec une meilleure flexibilité. La modulation OFDM est immune à la CD grâce à la présence d'un préfix cyclique *(cyclic prefix (CP))* et à la réduction de la complexité de l'égaliseur avec l'utilisation de symboles d'apprentissage *(training symbols (TS))*. De plus, l'OFDM offre un clair avantage en termes de flexibilité pour l'allocation de puissance par sous porteuse *(bit-power loading)* et la présence de symboles pilotes dans les sous porteuses en fonction des conditions du canal. L'OFDM multi-bande à détection cohérente MB-CO-OFDM *(Multiband-Coherent Optical-OFDM)* a donc été proposé en se basant sur les technologies récentes de convertisseurs numériques DAC et ADC et la réalisation possible dans les circuits intégrés ASIC ou FPGA.

0.2 Contexte du travail

Les systèmes CO-OFDM sont sensibles aux non linéarités présentes dans la chaine de traitement du signal comme le PAPR *(peak-to-average-power-ratio)*. Il sont également sensibles aux offsets de phase et de fréquence du LASER et aux erreurs de timing. Dans ces systèmes CO-OFDM, l'estimation et la compensation des imperfections sont réalisées dans le domaine numérique à l'aide d'algorithmes de traitement du signal. Avec la contrainte que chaque bande du système CO-OFDM travaille à un débit au Gb/s, les algorithmes de traitement du signal doivent être de faible complexité pour pouvoir supporter de telles fréquences de fonctionnement et d'échantillonnage et être implémentés dans des circuits comme des FPGA. De plus, les fréquences d'horloge des FPGA atteignent quelques centaines de MHz tandis que les fréquences d'échantillonnage des DAC/ADC requises sont de plusieurs GHz, ce qui implique des traitements hyper-parallèles au sein des architectures numériques de traitement. L'objectif de cette thèse est de proposer et d'implémenter des algorithmes parallèles performants et à faible complexité pour un transmetteur (émetteur + récepteur) CO-OFDM simple-bande et simple-polarisation, l'extension à plusieurs bandes et deux polarisations étant simple vu la *scalabilité* des algorithmes proposés.

L'approche que nous proposons pour réduire la complexité est tout d'abord d'utiliser

des algorithmes (e.g. synchronisation) à faible complexité avant de proposer des architectures parallèles efficaces pour leur implémentation matérielle. Les sections suivantes détaillent les résultats obtenus. Premièrement, une architecture et un algorithme à faible complexité sont proposés pour la synchronisation temporelle des trames et symboles OFDM. Deuxièmement, une architecture d'un transmetteur CO-OFDM complet est détaillé. Finalement, les architectures et algorithmes sont validés dans un contexte d'expérimentation *offline* puis temps réel.

0.3 Algorithme de synchronisation temporelle à faible complexité pour les systèmes OFDM

Un algorithme de synchronisation temporelle à faible complexité pour les systèmes OFDM est proposé dans un contexte de canaux sans fil sélectifs en fréquence. La proposition est basée sur une nouvelle séquence de symboles d'apprentissage basée sur les séquences de Chu modifiées (CAZAC) [10] définies par

$$a_k^{(r)} = \begin{cases} exp\left(i\frac{2\pi}{N_s}\left\lfloor\frac{rk^2}{2}\right\rfloor\right), & \text{pour } N_s \text{ pair} \quad (1) \\ exp\left(i\frac{2r\pi k(k+1)}{N_s}\right), & \text{pour } N_s \text{ impair} \quad (2) \end{cases}$$

où $0 \leq k < N_s$, $\gcd(r, N_s) = 1$ et $\lfloor a \rfloor$ dénotent la partie entière de a. Ici $r = 1$ est utilisé. La taille de l'alphabet est N_s pour la séquence de Chu modifiée en comparaison avec $2N_s$ pour la séquence de Chu. La nouvelle séquence de symboles d'apprentissage proposée dans cette thèse [11] est

$$[\boldsymbol{C}\ \boldsymbol{C}\ \boldsymbol{C}\ -\boldsymbol{C}], \boldsymbol{C} = [\boldsymbol{A}\ \boldsymbol{B}], \boldsymbol{B} = \boldsymbol{A}^*[-\boldsymbol{n}]$$

La partie $\boldsymbol{A}$ est construite en prenant la IFFT de la séquence de Chu modifiée [12] sur une taille $N_s = \frac{N}{8}$. Ensuite, $\boldsymbol{B}$ est construit à partir de $\boldsymbol{A}$ par un renversement temporel et une opération de conjugaison. Le motif de signes $[1\ 1\ 1\ -1]$ est conçu de façon à assurer une transition raide pour l'algorithme d'estimation grossière *(coarse)*. L'algorithme proposé contient trois étapes.

- Auto-corrélation initiale : l'opération d'auto-corrélation basée sur le délai est effectuée en utilisant un motif répétitif dans le symbole d'apprentissage. La métrique temporelle pour l'auto-corrélation initiale est donnée par

$$TM_{init}[n] = \left(\frac{L}{L-1} \cdot \frac{|P_{init}[n]|}{R_{init}[n]}\right)^2 \quad (3)$$

  ou P_{init} est la fonction d'auto-corrélation, R_{init} est la fonction d'énergie, TM_{init} est la métrique temporelle et L est le nombre de répétitions (ici $L = 4$) dans le symbole d'apprentissage proposé. Le terme $\frac{L}{(L-1)}$ est utilisé pour normaliser la valeur maximale à 1 au point de démarrage correct. Les expressions pour P_{init} et R_{init} sont

$$P_{init}[n] = \sum_{k=0}^{L-2} u[k] \sum_{m=0}^{M-1} r^*[n+kM+m] \cdot r[n+(k+1)M+m] \tag{4a}$$

$$R_{init}[n] = \sum_{k=0}^{L-1} \sum_{m=0}^{M-1} \left| r[n+kM+m] \right|^2 \tag{4b}$$

où $u[k] = p[k] \cdot p[k+1]$, $p[k]$ contient le motif de signes [1 1 1 −1], $k = 0, 1, ..., (L-1)$ et $M = N/L$. L'index de temps correspondant à la valeur maximale donne l'estimation temporelle initiale.

$$\widehat{\eta}_{init} = arg \max_n TM_{init}[n] \tag{5}$$

La figure 2.a trace $TM_{init}[n]$ pour un *Signal to Noise Ratio* (SNR) de 10 dB dans un canal sans fil sélectif en fréquence. L'algorithme d'estimation fine consiste en la correction d'un petit décalage pour trouver le point de démarrage correct.

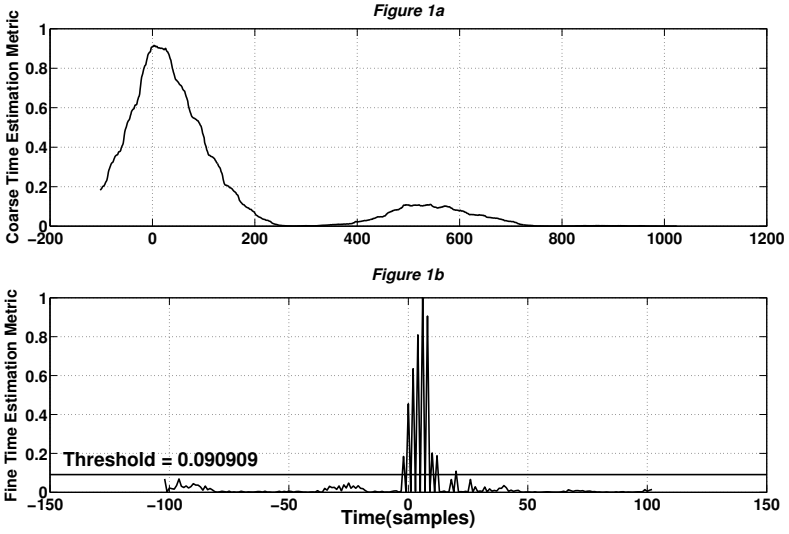

FIGURE 2: Tracé des fonctions de métriques temporelles grossière (a) et fine (b)

- Corrélation conjuguée symétrique : la métrique d'estimation temporelle fine est donnée par

$$TM_{fine}[n] = |P_{fine}[n]|^2 \tag{6}$$

avec

$$P_{fine}[n] = \sum_{k=0}^{\frac{N}{4}-1} r[n-k-1] \cdot r[n+k] - \sum_{k=\frac{N}{4}}^{\frac{N}{2}-1} r[n-k-1] \cdot r[n+k] \tag{7}$$

où TM_{fine} est la métrique temporelle fine et P_{fine} est l'opération de corrélation conjuguée symétrique. Le signe négatif pour $k \in [\frac{N}{4}, \frac{N}{2} - 1]$ est dû au motif de signes [1 1 1 −1], $n \in [-N_{cyp}, N_{cyp}]$. Cette intervalle pour n a été choisi pour que l'estimation initiale ne produise pas de pics en dehors de la longueur maximale du canal multi-trajets. La métrique de temps produit des pics qui sont proportionnels au carré des gains des trajets du canal. $Q[n]$ est calculé en normalisant toutes les

valeurs de $TM_{fine}[n]$ par la valeur maximale de $TM_{fine}[n]$:

$$Q[n] = \frac{TM_{fine}[n]}{max(TM_{fine}[n])} \tag{8}$$

La figure 2.b montre le tracé de $Q[n]$ pour un SNR de 10 dB. La figure montre les pics correspondant aux gains des trajets multiples. L'index temporel de la valeur maximale de $Q[n]$

$$\widehat{\eta}_{fine} = arg \max_{n}(Q[n]) \tag{9}$$

est utilisé comme point de démarrage pour la méthode des sommes fenêtrées.

- Sommation basée sur le seuil : les valeurs de $Q[n]$ sont limitées par un seuil de valeur β.

$$Q[n] = \begin{cases} Q[n], & Q[n] > \beta, \\ 0, & sinon, \end{cases} \tag{10}$$

β est le seuil qui sépare le signal de la composante de bruit dans $Q[n]$. Ce seuil est déterminé en utilisant la distribution de probabilité de la composante de bruit dans $Q[n]$. Les étapes sont les suivantes :

1. La séquence $Q[n]$ est passée dans l'algorithme de quantification de Lloyd-Max [13] utilisant trois niveaux de quantification.
2. Le niveau de quantification le plus bas est considéré comme du bruit. Il est observé que ce "cluster" de points suit une distribution log-normale. La moyenne (μ_n) et la variance (σ_n^2) du cluster de bruit sont calculées en premier. μ et σ pour une distribution log-normale sont

$$\mu = log\left(\frac{\mu_n^2}{\sqrt{\sigma_n^2 + \mu_n^2}}\right) \tag{11a}$$

$$\sigma = \sqrt{log\left(\frac{\sigma_n^2}{\mu_n^2} + 1\right)} \tag{11b}$$

3. Une vitesse constante de fausse alarme *(constant false alarm rate (CFAR))* de "α" est utilisée pour le calcul du seuil. L'équation est dérivée de l'intégrale de la fonction de distribution des probabilités *(probability distribution function)* du bruit dans l'intervalle [β, ∞].

$$\beta = e^{\left(\sqrt{2}\cdot\sigma\cdot erf^{-1}(1-2\cdot\alpha)+\mu\right)} \tag{12}$$

CFAR est utilisé pour toutes les valeurs de SNR. Une somme fenêtrée est calculée après avoir supprimé les valeurs de bruit en dessous du seuil β calculé.

$$E_p(n) = \sum_{k=0}^{S_w-1} Q(\widehat{\eta}_{fine} - n + k) \tag{13}$$

où S_w est la longueur de la fenêtre de sommation et J_m est la fenêtre de recherche pour la composante signal. Puis, le premier chemin d'arrivée est donné par

$$\widehat{\eta}_{first} = arg \max_{n} E_p(n) : n = 0, 1, \cdots J_m \tag{14}$$

Finalement,

$$\widehat{\eta}_{final} = \widehat{\eta}_{init} - \widehat{\eta}_{first} \tag{15}$$

indique l'estimation finale de l'index du début du symbole OFDM.

Résultats de simulation

La figure 3 montre l'erreur quadratique moyenne (MSE) de l'estimation temporelle dans un canal présentant des interférences entre symboles (ISI) pour différentes méthodes de synchronisation. Les méthodes basées sur la corrélation des délais (Schmidl, Minn, Shi) ont un MSE plus grand comparé aux méthodes utilisant des corrélations symétriques conjuguées (Park, Choi). La méthode proposée est meilleure que Park et est comparable à celle de Choi mais avec une complexité de calcul largement plus faible. Le nombre d'opérations sur des nombres réels en fonction de N est décrit dans la Table 1 pour différents algorithmes. Une réduction d'environ 80% de la complexité de calcul est obtenue pour la méthode proposée (pour $N_{sym} = 1126, N_{cyp} = 102$) par rapport à celles de Choi, Park et Zhou, tandis que les performances MSE restent très proches de celles de Choi.

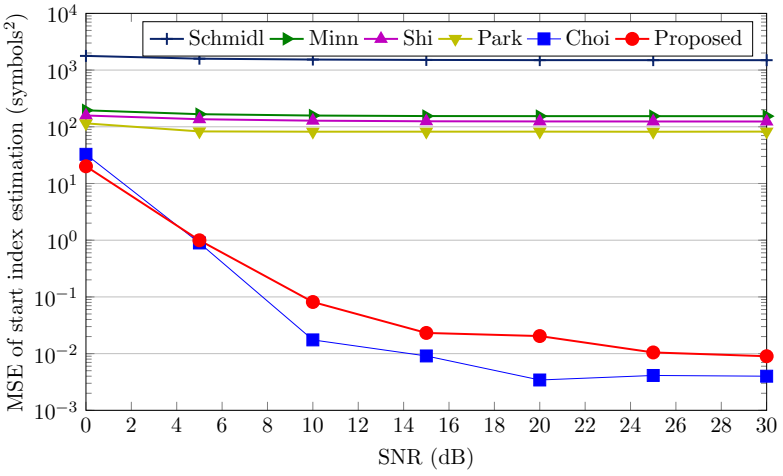

FIGURE 3: MSE de l'estimation temporelle en fonction du SNR dans un canal ISI

0.4 Synchronisation temporelle hiérarchique à faible complexité pour les systèmes CO-OFDM

La méthode synchronisation proposée dans la section précédente pour les systèmes sans-fil OFDM possède trois étapes pour atteindre de faible valeur de MSE :

TABLE 1: Nombre d'opérations réelles pour le calcul d'un point de la métrique temporelle

Algorithme	Multiplication	Addition	Division
Schmidl-Cox	15	13	1
Minn($L = 4$)	31	29	1
Shi	59	61	1
Park	$(2N + 11)$	$(2N + 7)$	1
Choi	$(2N + 7)$	$(2N + 3)$	1
Zhou	$(2N + 22)$	$(2N + 16)$	1
Algorithme proposé (coarse step) ($L = 4$)	31	29	1
Algorithme proposé (fine step)	$(2N + 3)$	$(2N - 1)$	1

- opération d'auto-corrélation (TM_{init}, Eq. 3),
- opération de corrélation conjuguée symétrique (TM_{fine}, Eq. 6),
- opération de sommation fenêtrée (E_p, Eq. 13).

Dans le cas de canaux optiques SMF, les valeurs de dispersion ne sont pas trop élevées et restent stables en comparaison avec l'effet multi-trajets des canaux sans fil qui peuvent présenter des retards très importants. Par conséquent, l'étape de sommation fenêtrée peut être éliminée et l'algorithme est réduit à ses deux premières étapes. Cet algorithme modifié est utilisé dans le cadre des canaux optiques SMF. Pour les comparaisons de performance, seuls les algorithmes basés sur l'auto-corrélation (Schmidl-Cox, Minn-Bhargava, Shi-Serpedin) sont reportés. En effet, les algorithmes de cross-corrélation (Choi, Park) sont trop complexes dans un contexte optique et ne génèrent pas de sorties à chaque cycle, comme requis dans ce contexte. Les étapes pour calculer le point de départ du symbole dans un système CO-OFDM $\eta_{final} = \eta_{init} - \eta_{fine}$ sont donc :

- opération d'auto-corrélation : l'équation 3 est utilisée sans modification pour le calcul de η_{init} ;
- opération de corrélation conjuguée symétrique : l'équation 6 est modifiée comme ci dessous pour réduire la complexité et la normalisation est effectuée en utilisant l'énergie du symbole.

$$\eta_{fine} = arg \max_{n} \left(TM^{lc}_{fine}[n] \right) \tag{16}$$

$$TM^{lc}_{fine}[n] = \frac{|P^{lc}_{fine}[n]|^2}{R^2_{fine}} \tag{17}$$

$$P^{lc}_{fine}[n] = \sum_{k=0}^{\frac{N}{4}-1} r[n-k-1] \cdot r[n+k] \tag{18}$$

$$R_{fine}[n] = \sum_{k=0}^{\frac{N}{4}-1} |r[n+k]|^2 \tag{19}$$

Résultats de simulation

La figure 4 trace le MSE de l'estimation temporelle dans un canal SMF avec un CFO de 4.75 pour les sous porteuses. On peut observer que l'algorithme proposé engendre une légère dégradation pour de faibles OSNR à cause de sa faible complexité dans le calcul de P_{fine}^{lc}. L'algorithme proposé donne par contre des améliorations significatives pour des OSNR plus élevés. Rappelons que dans tous les cas la complexité de calcul est largement réduite par rapport à l'état de l'art ce qui représente un grand avantage dans un contexte de communications optiques à très haut débit.

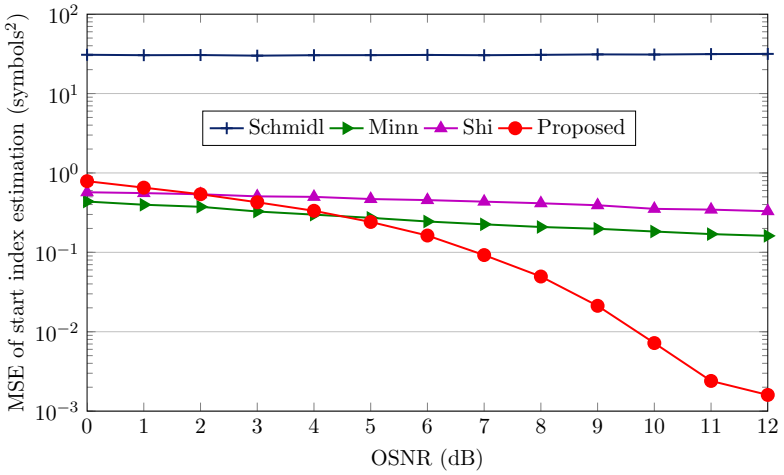

FIGURE 4: MSE de l'estimation temporelle en fonction du OSNR pour des canaux SMS et un CFO = 4.75

0.5 Architecture parallèle pour l'auto-corrélation

L'objectif de la parallélisation de l'algorithme d'auto-corrélation est d'obtenir une architecture "scalable" pour gérer de façon efficace des échantillons d'entrée multiples à chaque cycle d'horloge et atteindre le débit élevé désiré. Une architecture parallèle et évolutive est proposée dans ce travail et comparée aux travaux existants. L'architecture proposée posséde un parallélisme au niveau bloc et utilise à la fois la forme itérative et celle non-itérative du calcul de l'auto-corrélation pour atteindre un niveau de parallélisme suffisant. La forme non-itérative est utilisée pour initialiser les calculs tandis que la forme itérative calcule les point restant. Le choix de la taille des blocs permet de déterminer le partage des ressources. Soit l'algorithme d'auto-corrélation de Minn-Bhargava appliqué au symbole d'apprentissage $[A\ A\ A\ -A]$:

$$P_{mb}[n] = \sum_{k=0}^{L-2} p[k] \cdot p[k+1] \sum_{m=0}^{M_{mb}-1} r^*[n+m+kM_{mb}] \cdot r[n+m+(k+1)M_{mb}] \tag{20}$$

$$P_{mb}[n] = P_{mb}[n-1] + r^*[n] \cdot r[n+M_{mb}] - r^*[n+3M_{mb}] \cdot r[n+4M_{mb}] + 2r^*[n+2M_{mb}] \cdot r[n+3M_{mb}] \tag{21}$$

où Eq. 20 est la forme non itérative et Eq. 21 est la forme itérative, P_{mb} est la fonction d'auto-corrélation, M_{mb} est la taille de la partie répétitive (A), et $M_{mb} = \frac{N}{4}$. Si les équations sont réécrites pour un niveau de parallélisme $R = 4$ avec une taille de bloc M_{mb}, on obtient

$$\begin{aligned}
P_{mb}[n] &= \sum_{k=0}^{L-2} p[k] \cdot p[k+1] \sum_{m=0}^{M_{mb}-1} r^*[n+m+kM_{mb}] \\
P_{mb}[n+M_{mb}] &= \sum_{k=0}^{L-2} p[k] \cdot p[k+1] \sum_{m=0}^{M_{mb}-1} r^*[n+m+(k+1)M_{mb}] \\
P_{mb}[n+2M_{mb}] &= \sum_{k=0}^{L-2} p[k] \cdot p[k+1] \sum_{m=0}^{M_{mb}-1} r^*[n+m+(k+2)M_{mb}] \\
P_{mb}[n+3M_{mb}] &= \sum_{k=0}^{L-2} p[k] \cdot p[k+1] \sum_{m=0}^{M_{mb}-1} r^*[n+m+(k+3)M_{mb}]
\end{aligned} \tag{22}$$

$$\begin{aligned}
P_{mb}[n+1] &= P_{mb}[n] - r^*[n] \cdot r[n+M_{mb}] - r^*[n+3M_{mb}] \cdot r[n+4M_{mb}] \\
&\quad + 2r^*[n+2M_{mb}] \cdot r[n+3M_{mb}] \\
P_{mb}[n+1] &= P_{mb}[n] - r^*[n] \cdot r[n+M_{mb}] - r^*[n+3M_{mb}] \cdot r[n+4M_{mb}] \\
&\quad + 2r^*[n+2M_{mb}] \cdot r[n+3M_{mb}] \\
P_{mb}[n+1] &= P_{mb}[n] - r^*[n] \cdot r[n+M_{mb}] - r^*[n+3M_{mb}] \cdot r[n+4M_{mb}] \\
&\quad + 2r^*[n+2M_{mb}] \cdot r[n+3M_{mb}] \\
P_{mb}[n+1] &= P_{mb}[n] - r^*[n] \cdot r[n+M_{mb}] - r^*[n+3M_{mb}] \cdot r[n+4M_{mb}] \\
&\quad + 2r^*[n+2M_{mb}] \cdot r[n+3M_{mb}]
\end{aligned} \tag{23}$$

0.5.1 Architecture parallèle partielle(PSBP)

Dans cette première version d'une architecture parallèle, les ressources de calcul sont partagées entre les calculs itératifs et non itératifs. L'architecture possède donc deux modes d'opération, un mode non-itératif pour l'initialisation du calcul et un second mode itératif pour le reste des calculs. Soit le calcul de R M_{mb} sorties de l'auto-corrélation avec une architecture PSBP incluant R blocs en parallèle, chaque bloc calcule $\frac{R\,M_{mb}}{R}$ sorties. L'ordre des calculs est le suivant :

- R points initiaux sont calculés en mode non itératif ce qui nécessite M_{mb} cycles pour l'algorithme MBA,
- les $(M_{mb} - 1)$ points restant sont calculés en mode itératif.

Après le calcul de R M_{mb} sorties, le même processus est répété pour les R M_{mb} sorties suivantes. L'architecture prend $(2M_{mb} - 1)$ cycles pour le calcul de M_{mb} points d'auto-corrélation par bloc. L'architecture est appelée "parallèle partielle" car elle ne produit pas de sortie à chaque cycle et elle possède un délai équivalent à la partie initialisation. Cependant le nombre de ressources est plus faible que dans la proposition suivante.

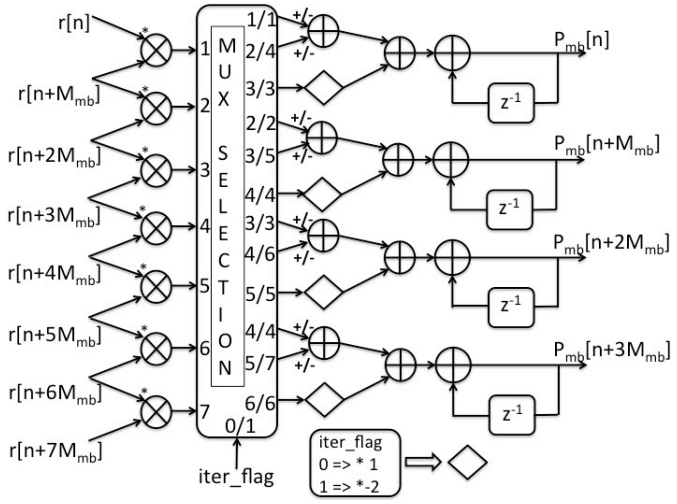

FIGURE 5: Architecture PSPB proposée pour le calcul de P_{mb} avec MBA

La table 2 présente la complexité architecturale en fonction de R pour l'algorithme de Minn-Bhargava. Elle est comparée avec la proposition de Kaneda pour l'algorithme de Schmidl-Cox. Notre architecture requiert 12 multiplieurs de plus que celle de Kaneda du fait de la plus grande complexité de l'algorithme considéré. Mais la surface totale reste plus faible en particulier au nombre réduit d'additionneurs.

TABLE 2: Complexité architecturale en fonction de R

Architecture proposée pour Minn-Bhargava

Algorithm	Real Multipliers	Real Adders
P_{mb}	$4(R+3)$	$2(3R+3)$

Architecture de Kaneda pour Schmidl-Cox

Algorithm	Real Multipliers	Real Adders
P_{sc}	$4R$	$2R(M_{sc}/2+1)$

0.5.2 Architecture parallèle complète (FSBP)

L'architecture proposée dans la section 0.5.1 utilise M_{mb} cycles pour initialiser les calculs ce qui induit un délai à chaque nouveau calcul et implique donc des ressources mémoire supplémentaires pour le stockage des symboles OFDM ainsi qu'un non respect des contraintes temps réel. L'architecture est donc modifiée pour éviter ces défauts et obtenir une architecture complètement parallèle capable de produire M_{mb} sortie d'auto-corrélation en M_{mb} cycles. La modification proposée consiste en l'ajout d'un bloc pour le calcul des points initiaux en parallèle de façon àce que l'auto-corrélation produise un calcul à cycle. Une

architecture parallèle à $R = 4$ blocs pour MBA est présentée à la figure 6. La table 3 donne la complexité architecturale pour le calcul de P_{mb}. Comparée à nouveau à celle de Kaneda [14], des réductions en termes de surface de 17 à 72% sont obtenus en fonction de la taille du symbole.

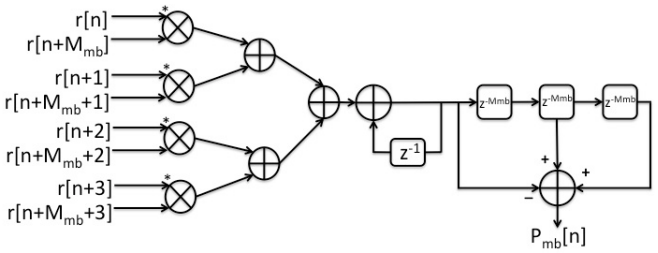

FIGURE 6: Architecture parallèle FSPB proposée pour le calcul de P_{mb} avec MBA et $R = 4$

TABLE 3: Complexité architecturale en fonction de R

Algorithm	Real Multipliers	Real Adders
P_{mb} (Initial Point)	$4R$	$4R$
P_{mb} (Iterative Point)	$4(R+3)$	$2(3R+3)$
P_{mb} (Total)	$8R+12$	$10R+6$

0.6 Architecture parallèle pour les systèmes CO-OFDM

Dans cette section une architecture parallèle d'émetteur-récepteur pour un transmetteur CO-OFDM est proposée.

0.6.1 Emetteur

A l'émission, la IFFT est le bloc principal en termes de complexité. Le choix du radix utilisé pour le FFT est donc crucial et peut donc influencer la complexité de l'architecture. Pour $N = 256$ la complexité en millions d'opérations par seconde (MOPS) est calculée pour des FFT radix-$2/4/2^2$ et split-radix et reportée dans la table 4 pour supporter un débit de 7.3 Gbit/s. Ensuite, le nombre total d'opérations pour supporter un débit total $D_{b,total} \geq 100$ Gb/s est reporté dans la dernière colonne de la table 4. Ces résultats montrent qu'un gain de 800 GOPS peut être obtenu pour les algorithmes radix-$4/2^2$ par rapport au radix-2, tandis que 200 GOPS supplémentaires peuvent être atteints par split-radix. Nous avons retenu le radix-2^2 car sa complexité architecturale est plus faible.

0.6.2 Récepteur

Les algorithmes utilisés pour la synchronisation temps, l'estimation du CFO, la FFT, l'estimation du canal, l'égalisation, l'estimation de l'erreur en phase et la compensation

TABLE 4: Complexité algorithmique pour le calcul de N sorties de la IFFT avec N=256

Radix Used	Real Multipli-cations	Real Additions	Total Operations	GOPS (D_b) for 7.3 Gb/s	TOPS ($D_{b,total}$) for 117 Gb/s
Radix-2	4096	6144	10240	294.4	4.7
Radix-4	3072	5632	8704	248.2	3.9
Radix-2^2	3072	5632	8704	248.2	3.9
Split-Radix	2731	5462	8193	233.6	3.7

sont décrits dans cette section. La complexité algorithmique pour le calcul de N sorties et pour un système à 117 Gb/s y est aussi présentée. Des optimisations spécifiques sur le format des données sont réalisées pour réduire la complexité.

- Synchronisation temporelle grossière - l'algorithme utilisé est celui proposé dans cette thèse. Sa complexité pour le calcul de N sorties est donnée dans la table 5, pour une sous bande et pour un débit des sorties de 117 Gb/s. Le CFO est estimé par auto-corrélation au point de départ du symbole.

$$\begin{aligned} P_{mb}[n+1] = P_{mb}[n] &- r^*[n] \cdot r[n+M_{mb} \\ &- r^*[n+3M_{mb} \cdot r[n+4M_{mb}] \\ &+ 2 \cdot r^*[n+2M_{mb}] \cdot r[n+3M_{mb}] \end{aligned} \tag{24}$$

  où P_{mb} est l'auto-corrélation, M_{mb} est la longueur du symbole d'apprentissage utilisé $[A\ A\ A\ -A]$.

TABLE 5: Complexité algorithmique (auto-corrélation) de notre proposition

Algo. Used	Real Multipli-cations	Real Additions	Total Operations	GOPS (D_b) per sub-band	TOPS ($D_{b,total}$) for 117 Gb/s
Minn -Bhargava ($L=4$)	3072	3072	6144	175.2	2.8

- FFT - Une FFT Radix-2^2 est choisie et implémentée.
- Estimation CFO entière - Une cross-corrélation avec une séquence connue de symboles est réalisée pour l'estimation CFO. Soit une séquence connue de longueur $N_{ifo} = N/4$ notée $z[n]$. Comme, l'estimation CFO est faite dans le domaine de fréquences la séquence connue peut être construite à partir de symboles QPSK de valeur $(\pm 1 \pm 1j)$. L'opération de cross-corrélation devient

$$M_{ifo}[n] = |P_{ifo}[n]|^2 \tag{25}$$

$$P_{ifo}[n] = \sum_{m=0}^{N_{ifo}-1} r[n+m] \cdot z^*[n+m] \tag{26}$$

où n est l'index de recherche, $n \in [-W_s, .., -2, 0, 2, .., W_s]$, où W_s est l'index maximum. Ici $W_s = 20$ est choisi comme valeur maximale. La valeur de N_{ifo} est fixée à 32. La complexité algorithmique est reportée dans la table 6. Grâce à l'utilisation d'une constellation QPSK, $(\pm 1 \pm j)$, les multiplications complexes peuvent être complètement éliminées et la complexité ainsi réduite. Un gain de 39.8 MOPS est ainsi obtenu.

TABLE 6: Complexité algorithmique de l'estimation CFO entière

Algo. Used	Real Multipli-cations	Real Add-itions	Total Oper-ations	MOPS (D_b) per sub-band	MOPS ($D_{b,total}$) for 100Gb/s
IFO Estimation Non-Optimized	15360	7680	23040	3.68	59
IFO Estimation Optimized	82	2624	2706	1.3	20.8

- Estimation du canal et égalisation - Les algorithmes des moindres-carrés (LS) et LS normalisés sont utilisés pour l'estimation du canal. La complexité algorithmique est reportée dans la table 7. A nouveau, la complexité de la méthode LS peut être réduite en utilisant le symbole $(\pm 1 \pm j)$. Cette optimisation reste valable pour les modes à simple et double polarisation. Des gains de 29.2 GOPS pour LS et de 21.9 GOPS pour NLMS sont atteints.

- Estimation CPE et compensation - Une estimation CPE basée sur des symboles pilotes [15] est réalisée pour l'estimation du bruit de phase du LASER. La complexité algorithmique est reportée dans la table 8.

Le bilan final montre un gain en complexité de plus 800 GOPS par rapport à une implémentation non optimisée.

0.7 Experimentations

Des expérimentations temps réel et *off-line* ont finalement été réalisées dans cette thèse pour valider les paramètres du système OFDM dans un contexte de communications optiques. Tout d'abord, des scénarios en temps différé *(off-line)* ont été conduits à l'aide d'un générateur de signaux AWG comme émetteur et d'un oscilloscope numérique rapide (DSO) comme récepteur. La figure 7 montre la configuration hétérodyne dans laquelle des sources LASER distinctes sont utilisées à l'émission et à la réception.

- Emetteur Electro-Optique - La fréquence porteuse de l'émetteur et générée par un LASER à cavité externe (ECL) àune longueur d'ondes de 1540 nm. Le signal optique est amplifiée avec une fibre PMF *(polarization maintaining fiber)* qui amplifie le

TABLE 7: Complexité algorithmique de l'estimation de canal

Algo. Used	Real Multipli-cations	Real Add-itions	Total Oper-ations	GOPS (D_b) per sub-band	TOPS ($D_{b,total}$) for 117 Gb/s
LS Single Pol. Non-Optimized	2048	768	2816	80.3	1.28
LS Single Pol. Optimized	1024	768	1792	51.1	0.81
LS Dual Pol. Non-Optimized	8192	3072	11264	321.2	5.1
LS Dual Pol. Optimized	4096	3072	7168	204.4	3.2
NLMS Single Pol. Non-Optimized	2816	2048	4864	138.7	2.2
NLMS Single Pol. Optimized	2304	1792	4096	116.8	1.8
NLMS Dual Pol. Non-Optimized	5632	4096	9728	277.4	4.4
NLMS Dual Pol. Optimized	4608	3584	8192	233.6	3.7

TABLE 8: Complexité algorithmique de l'estimation et de la compensation CPE

Algo. Used	Real Multipli-cations	Real Add-itions	Total Oper-ations	GOPS (D_b) per sub-band	GOPS ($D_{b,total}$) for 100Gb/s
CPE Estimation Optimized	1024	512	1536	43.8	700.8

signal de 15 dBm. Un coupleur 3 dB divise la puissance pour attaquer le contrôleur de polarisation de façon à maximiser la puissance optique sur le modulateur optique MZM (Mach-Zender Modulator). La sortie du contrôleur avec une puissance de 11 dBm sert de porteuse au bloc MZM. Le signal, généré à partir du logiciel Matlab, est envoyé à l'AWG et au driver RF.

- Récepteur Opto-Electronique - Un atténuateur optique est utilisé pour faire varier le rapport signal à bruit *(optical signal-to-noise ratio (OSNR))*. Un filtre passe-bande optique (BPF) sélectionne la bande passante autour de la fréquence porteuse dans l'intervalle des 3 nm. L'atténuateur permet de régler la puissance du signal optique à l'entrée du détecteur cohérent et la valeur maximale du signal est fixée à -5 dBm. Le

signal après détection cohérente est envoyé à l'oscilloscope pour échantillonnage et stockage. Le signal ainsi échantillonné peut être traité en temps différé sur le logiciel Matlab.

La figure 8 montre les performances en termes de taux d'erreurs binaires (BER) de cette expérimentation ce qui permet de valider les différents algorithmes du transmetteur.

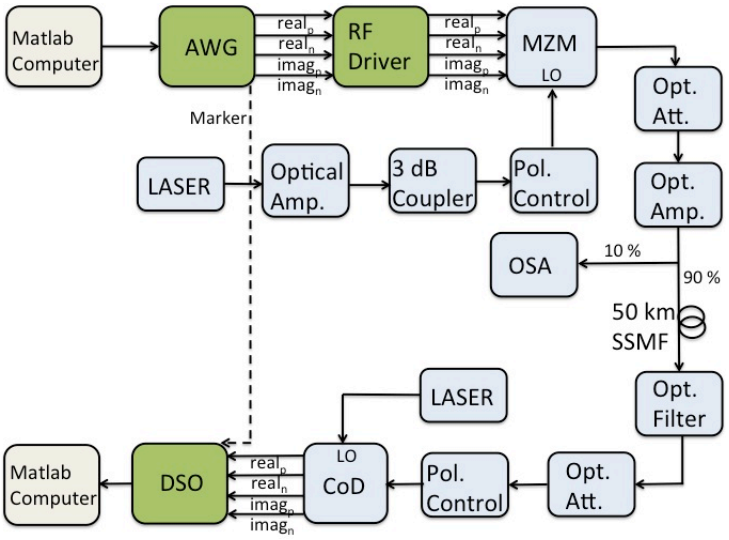

FIGURE 7: Configuration hétérodyne à détection cohérente avec une fibre SSMFde 50 km

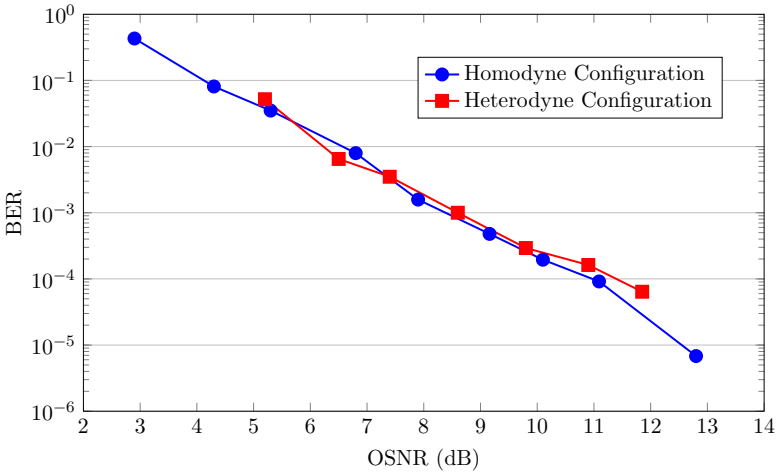

FIGURE 8: BER vs SNR pour un système CO-OFDM simple bande hétérodyne

Dans un second temps, des expérimentations "temps-réel" sont réalisées et reportées. Elles utilisent une plateforme temps-réel FPGA développée dans le cadre du projet FUI 100GFLEX. Cette plateforme intègre des FPGA Xilinx et Altera associés à des DAC et ADC rapides. Les différents blocs implémentés sont décrits ci dessous. Les figures 9 et 10 montrent les plateformes d'émission et de réception respectivement.

Une expérimentation *back-to-back* électrique sans CFO permit de valider les algorithmes de synchronisation de trame OFDM. L'architecture proposée pour l'algorithme

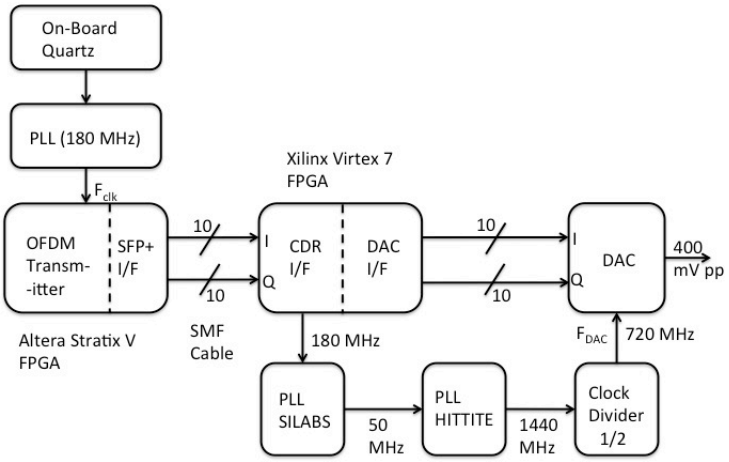

FIGURE 9: Plateforme FPGA temps réel d'émission

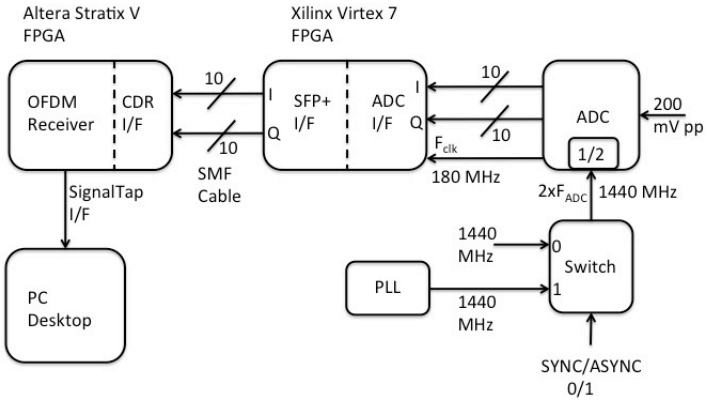

FIGURE 10: Plateforme FPGA temps réel de réception

de Minn-Bhargava est utilisée pour l'estimation du point de départ de la trame. L'architecture est synthétisée à partir du langage C en utilisant l'outil de synthèse de haut niveau HLS CatapultC. Les résultats reportés dans la thèse valident les algorithmes et démontrent l'efficacité de l'architecture proposée en termes de performance BER et de surface.

0.8 Conclusion

Les systèmes de communications optiques à très haut débit sont construits à partir des techniques de pointe pour la détection, la modulation et la compensation de dispersion tels que, la détection cohérente, les modulations multi-porteuses orthogonales (OFDM) et la compensation électronique des dispersions (EDC). La réapparition de la détection cohérente dans les systèmes de communication optique a été rendue notamment possible par les progrès dans les circuits numériques dans les technologies avancées. La détection cohérente possède une meilleure sensibilité pour la détection du signal par rapport aux méthodes de détection directe. Elle permet d'utiliser des transmissions à double polarisation et conserve les informations de phase du signal optique et les transfert dans le

domaine électrique. L'utilisation de la modulation OFDM fournit une flexibilité significative et l'utilisation efficace de la bande passante allouée. En raison de la disponibilité des informations de phase dans le domaine numérique, les processeurs DSP de faible coût peuvent être utilisés pour la compensation des dispersions dans le domaine numérique qui rend la solution flexible et reconfigurable. Mais, l'introduction du système CO-OFDM *(Coherent-Optical OFDM)* à la place de système de IM-DD *(Intensity Modulation-Direct Detection)* augmente significativement le coût du système avec un plus grand nombre de composants optiques et une quantité plus élevée de ressources électroniques requises pour la réception du signal. À l'heure actuelle, cela rend cette solution uniquement justifiable pour des transmissions à longue portée, même si le nombre de ressources par rapport à un système mono-porteuse à détection cohérente et modulation à quatre états (DP-CO-QPSK). Le choix de l'algorithme et l'optimisation de la précision des calculs en virgule fixe de l'architecture peuvent réduire de façon significative les ressources nécessaires pour la réalisation de systèmes CO-OFDM.

Dans cette thèse, des algorithmes à faible complexité et des architectures parallèles et efficaces sont explorés pour les systèmes CO-OFDM. Tout d'abord, des algorithmes de faible complexité pour la synchronisation et l'estimation du décalage en fréquence en présence d'un canal dispersif sont étudiés. Un nouvel algorithme de synchronisation temporelle à faible complexité qui peut résister à grande quantité de retard dispersif est proposé et comparé par rapport aux propositions antérieures. Ensuite, le problème de la réalisation d'une architecture parallèle à faible coût est étudié et une architecture parallèle générique et évolutive qui peut être utilisée pour réaliser tout type d'algorithme d'auto-corrélation est proposé. Cette architecture est ensuite étendue pour gérer plusieurs échantillons issus du convertisseur analogique/numérique (ADC) en parallèle et fournir une sortie qui suive la fréquence des ADC. L'évolutivité de l'architecture pour un nombre plus élevé de sorties en parallèle et les différents types d'algorithmes d'auto-corrélation sont explorés.

Une approche d'adéquation algorithme-architecture est ensuite appliquée à l'ensemble de la chaîne de l'émetteur-récepteur CO-OFDM. Du côté de l'émetteur, un algorithme IFFT à *radix*-2^2 est choisi pour et une architecture parallèle Multipath Delay Commutator (MDC) Feed-forward (FF) est choisie car elle consomme moins de ressources par rapport aux architectures MDC-FF en *radix*-2/4. Au niveau du récepteur, un algorithme efficace pour l'estimation du *Integer CFO* est adopté et implémenté de façon optimisée sans l'utilisation de multiplicateurs complexes. Une réduction de la complexité matérielle est obtenue grâce à la conception d'architectures efficaces pour la synchronisation temporelle, la FFT et l'estimation du CFO. Une exploration du compromis entre la précision des calculs en virgule fixe et la complexité du matériel est réalisée pour la chaîne complète de l'émetteur-récepteur, de façon à trouver des points de fonctionnement qui n'affectent pas le taux d'erreur binaire (TEB) de manière significative. Les algorithmes proposés sont validés à l'aide d'une part d'expériences *off-line* en utilisant un générateur AWG *(arbitrary waveform generator)* à l'émetteur et un oscilloscope numérique à mémoire (DSO) en sortie de la détection cohérente au récepteur, et d'autre part un émetteur-récepteur temps-réel

basé sur des plateformes FPGA et des convertisseurs numériques. Le TEB est utilisé pour montrer la validité du système intégré et en donner les performances.

Chapter 1

Introduction

In today's world, the demand of Internet bandwidth is increasing at an exponential rate, compared to increase in demand in the previous decade. One of the reasons being the arrival of always connected devices like smartphones, tablets, and the emergence of video based applications like YouTube, the Web moving towards richer interactive applications. With more and more people connected to the Internet, it creates huge burden on all the nodes of the communication network. Figure 1.1 shows the predicted increase in monthly traffic for the next five years by application type. It can be seen that video-based applications will continue to increase in number and put enormous pressure on the underlying network.

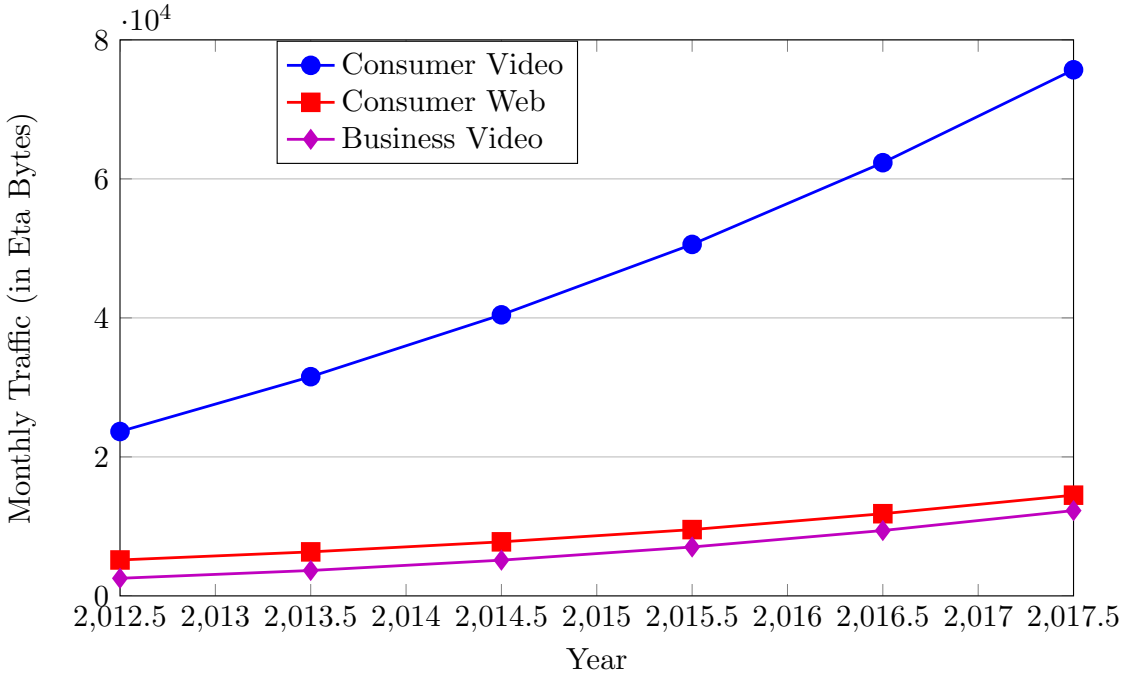

FIGURE 1.1: Cisco Visual Networking Index (VNI) Prediction of growth of internet by Application Type (Updated May 2013). The ordinate units is in Eta Bytes (EB). Total traffic is 2017 is predicted to be three times larger than 2012 [1].

The Internet is built upon many communication standards, which use different types of physical medium to communicate data bits around the world. One of the most important physical medium which forms the backbone of the network is Optical Communication system. Optical Communication system carries data presently over very long distances (Submarine networks, Long-haul networks), medium distances (Metro networks, Access Networks). With the introduction of Fiber-to-the-Home (FTTH), optical fiber communication system is also serving end users directly. Submarine networks are undersea networks which have links supporting distances of more than 2000 km. The terrestrial optical communication network can be divided into three major types. Figure 1.2 shows these three types of network which are classified as a function of distance. Core Network (CN)

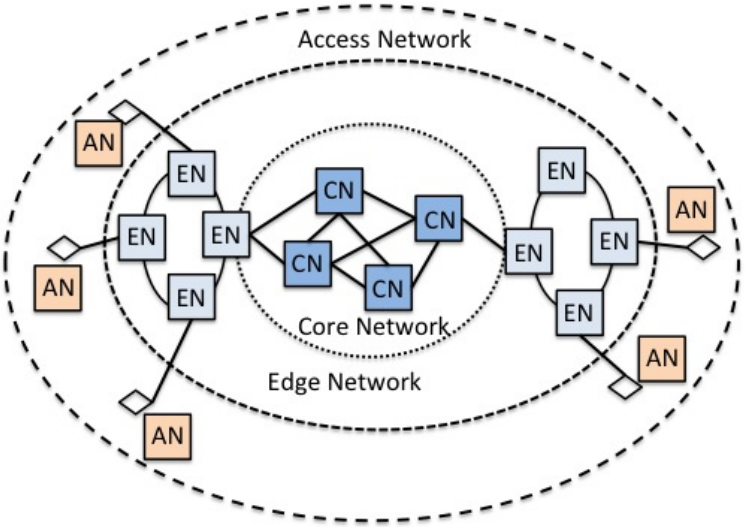

FIGURE 1.2: Typical Optical Network Architecture, CN - Core Node, EN - Edge Node, AN - Access Node

is a long-haul interconnection network that covers hundreds/thousands of kilometers connecting large cities, countries and even continents. It uses mesh topology. Optical Edge Network (EN) connects smaller geographical areas, covering distances of tens of kilometers, which is commonly known as known as Metropolitan Area Network (MAN).The Access Network (AN) is the peripheral part of the optical network, which is commonly known as Local Area Network (LAN). It uses star topology. Presently, the optical network uses 10/40 Gb/s single carrier modulation scheme in a single band for the data transmission.

Optical communication system uses optical fiber as the primary medium for transmission. Optical fiber has low attenuation coefficient ($\alpha_{dB} \leq 0.35\ dB/km$) in the wavelength range of 1300 nm to 1700 nm. It offers huge amount of bandwidth for transmission. Coupled with Erbium-Doped Fiber Amplifier (EDFA) technology, which amplifies optical signals in the wavelength range of 1530 nm - 1600 nm, optical communication system can reach thousands of kilometers. Total data rate is increased in optical communication systems by the use of Wavelength Division Multiplexing (WDM) techniques and Dense Wavelength Division Multiplexing (DWDM). The bandwidth (1530 nm - 1565 nm) used

for transmission is called C-band. 1 Tb/s is the maximum capacity in C-band using 10 Gb/s DWDM Long-haul transmission network, which uses Non Return-to-Zero (NRZ)-On-Off Keying (OOK) modulation for transmission.

To realize still higher speeds of transmission, data rate on individual channels have to be increased from 10 Gb/s to 100 Gb/s [6]. Increasing the symbol rate of 10 Gb/s OOK transmissions is simply not a viable solution because of dispersion effects in the optical fiber. Chromatic Dispersion (CD) causes Inter Symbol Interference (ISI) at very high symbol rates and hence severely impacts single-carrier transmission. At the receiver, the complexity of time-domain equalizer increases significantly with increased symbol rate. Also, Polarization Mode Dispersion (PMD) effects are more severe at very high data rates. Compensation of PMD is done by using bulky rotators which is not flexible. Hence, compensation of both of these effects is challenging and solutions are not cost-effective. So, the Intensity Modulation-Direct Detection (IM-DD) NRZ-OOK system cannot be scaled to 100 Gb/s data rates per channel.

To realize higher speeds, Coherent Detection (CoD) has been reintroduced into optical communication system. Direct Detection (DD) was preferred over CoD because of its simplicity in complexity and cost. CoD has come back into prominence due to advancements in VLSI circuits. CoD [7][8][9] offers additional advantages of higher detection sensitivity, higher symbol rates, use of dual polarization and more importantly the amplitude and phase information is conserved when crossing from optical to electrical domain. This opens up the possibility of Electronic Dispersion Compensation (EDC) using Digital Signal Processing (DSP) algorithms, which are low cost, powerful and reprogrammable. This has led to development of Coherent Optical Dual Polarization QPSK (CO-DP-QPSK) systems which can work at 100 Gb/s. These systems use dual polarization and two bits per symbol to essentially deliver four times the bit rate that allows the DSP to operate at four times the lower frequency. Since it uses single-carrier scheme, it requires Finite Impulse Response (FIR) filter for equalization. Also, the CO-DP-QPSK adopted in the 100 Gb/s standard uses blind Channel Estimation (CE), which increases estimation complexity [16]. With the use of Quadrature Phase Shift Keying (QPSK), the use of Digital-to-Analog Converter (DAC) can be avoided now [16]. But, if in future, higher modulation format is adapted, DAC will have to be used and this will increase transmitter and equalizer complexity [16].

In the same time, Coherent Optical-OFDM (CO-OFDM)[17][18] has been proposed as a possible candidate for transmission for 100 Gb/s/400 Gb/s data rate and beyond. CO-OFDM as the name indicates combines the technique of coherent detection (CoD) and multi-carrier modulation of Orthogonal Frequency Division Multiplexing (OFDM) [19] to counter the optical channel. OFDM is inherently immune to CD due to presence of Cyclic Prefix (CP) and with the usage of Training Symbols (TS), the equalizer complexity can be reduced significantly to a one-tap equalizer. Also, OFDM offers all the flexibility advantages of allocation of power per sub-carrier (bit-power loading), pilot sub-carrier locations based on channel conditions.

Presently Ethernet is used at a line rate of 10 Gb/s. Due to large presence of Ethernet, future increases in line rate will want to use Ethernet standard and change only the technology to support higher line rates. The next upgrade step is 100 Gb/s Ethernet. The jump from 10 Gb/s Ethernet to 100 Gb/s is necessary because router-to-router trunk connectivity has already reached 100 Gb/s [20] and also achieving line rate of 100 Gb/s compared to 10 lines of 10 Gb/s results in cost reduction per Gb/s. This makes achieving 100 Gigabit Ethernet (100GbE) a very important milestone to support the present day demands. Along with the present goal of 100 GbE and towards a future goal of 1 Tb/s Ethernet (1TbE), the solutions adopted should have these desirable properties, which can make the solution future proof.

- They should be compatible with the present optical infrastructure which comprises of single mode fiber having varying range of CD, Dispersion Compensation Fiber (DCF) and other types of fiber.
- They should be scalable to higher speeds easily and can support reconfigurable networks which manages bandwidth at a higher software level.

1.1 Context of the Work

With traffic on the Internet growing exponentially and management of quality-of-service (QoS) requiring flexibility at all levels of hierarchy of the optical communication system. The presence of WDM and Reconfigurable Optical Add Drop Multiplexer (ROADM) gives flexibility to cope up with dynamic changes in bandwidth requirement in different parts of network. But, with required bandwidth moving towards 100 Gb/s between routers, it has become necessary for the network to be dynamically reconfigurable and software controlled. Also, the physical layer has to become transparent to different types of network it is traversing through. As the speed is increased to 100 Gb/s, the signal loses its immunity to CD, PMD and ROADM filtering effects. With the presence of different types of optical fibers having large range of CD and PMD values, it is very difficult to maintain CD and PMD compensation using lossy, bulky optical compensators. With these set of flexibility and inherent immunity to dispersion requirements of optical networks, it is important to choose a solution, which offers these features without significant costs.

CO-OFDM has inherent advantages of being immune to dispersion mechanisms and also is extremely flexible, e.g. using different data mapping schemes on sub-carriers (bit-loading), pilot insertion, etc. Also, with the use of Multiband-OFDM (MB-OFDM), the pressure on DAC/ADC is reduced and it allows the present day solution of multiple sub-banded OFDM systems to reach the total data rate of 100 Gb/s. But, CO-OFDM has some disadvantages, which put the brakes on scaling the system efficiently. CO-OFDM system is sensitive to non-linearities in the optical communication chain and also it has a high Peak to Average Power Ratio (PAPR). CO-OFDM system is sensitive to frequency and phase offsets of the LASER in the system and also to timing offset. Also, all the

estimation and compensation of non-idealities are done in digital domain, which makes the algorithms complex and has to be adapted to optical system taking from Wireless OFDM domain. These algorithms now have to operate at much higher speeds compared to Wireless OFDM implementation and have to be easily scalable as well.

With FPGA maximum clock frequencies much lower than DAC/ADC frequencies, it forces every DSP block to be parallelized or simply replicated to match the input data rate. Simply replicating the blocks results in huge amount of area and is not feasible in the long term. In this thesis, the goal is to have a fully scalable parallel CO-OFDM system that can support a very-high data rate. At the transmitter, Inverse Fast Fourier Transform (IFFT) is the major component, which needs to be parallelized and it needs to be parallelized efficiently. The choice of radix and number of parallel outputs is explored to get the best area efficient design. CO-OFDM transmitter is a feed-forward system, while CO-OFDM receiver is a system with a feedback loop. Also, it consists of time, frequency, phase offset estimation and compensation blocks which needs to be efficiently parallelized along with Fast Fourier Transform (FFT) which forms the major block of the receiver. Fixed-point analysis of the complete OFDM chain is done which helps in reduction of the area. All the analysis is done for a single polarization CO-OFDM system. It can be extended to dual polarization CO-OFDM system by the inclusion of Multiple Input Multiple Output (MIMO) block which can separate the two polarization components and feed into corresponding chains.

1.2 Contributions

The contributions of this work are given as follows.

1. A new low-complexity hierarchical coarse time synchronization algorithm is proposed. For an OFDM signal in the presence of a multi-path channel, initial time synchronization is estimated by using hierarchical approach of auto-correlation and cross-correlation. It also gives fractional Carrier Frequency Offset (CFO) estimate.

2. A scalable and parallel architecture for initial time synchronization algorithm is proposed. The proposed architecture is required to perform real-time synchronization when the ADC sampling frequency is much higher than FPGA operating frequency.

3. A complete parallel architecture for all the blocks of the CO-OFDM transceiver is proposed. The scalability towards 100 Gb/s is detailed. The scalability of individual algorithms is explored in detail. A complete fixed-point analysis of the proposed parallel CO-OFDM system is done. Area reduction due to the analysis is reported.

4. A High-Level Synthesis (HLS) approach to designing of the CO-OFDM blocks is taken, which reduces design time and helps in Design Space Exploration of the DSP blocks.

5. Validation of the proposed algorithms/architecture is done in a practical set-up using offline and online experiments. Offline experiments are conducted in Optical Laboratory setup and online experiments using real-time FPGA platform developed as part of FUI 100GFLEX project.

Figure 1.3 shows the possible power saving opportunities at different stages of VLSI design flow. Resource savings also follows a similar trend. As can be observed, low-complexity algorithm and architecture saves resources at block level like savings of multipliers, adders which down the flow results in significant savings compared to savings obtained at Register Transfer (RT) or Logic Level. This approach is taken in this Thesis to optimize the resource consumption in the search of parallel CO-OFDM transceiver algorithms/architectures and it is expressed using high-level synthesis (HLS) language of CatapultC [21].

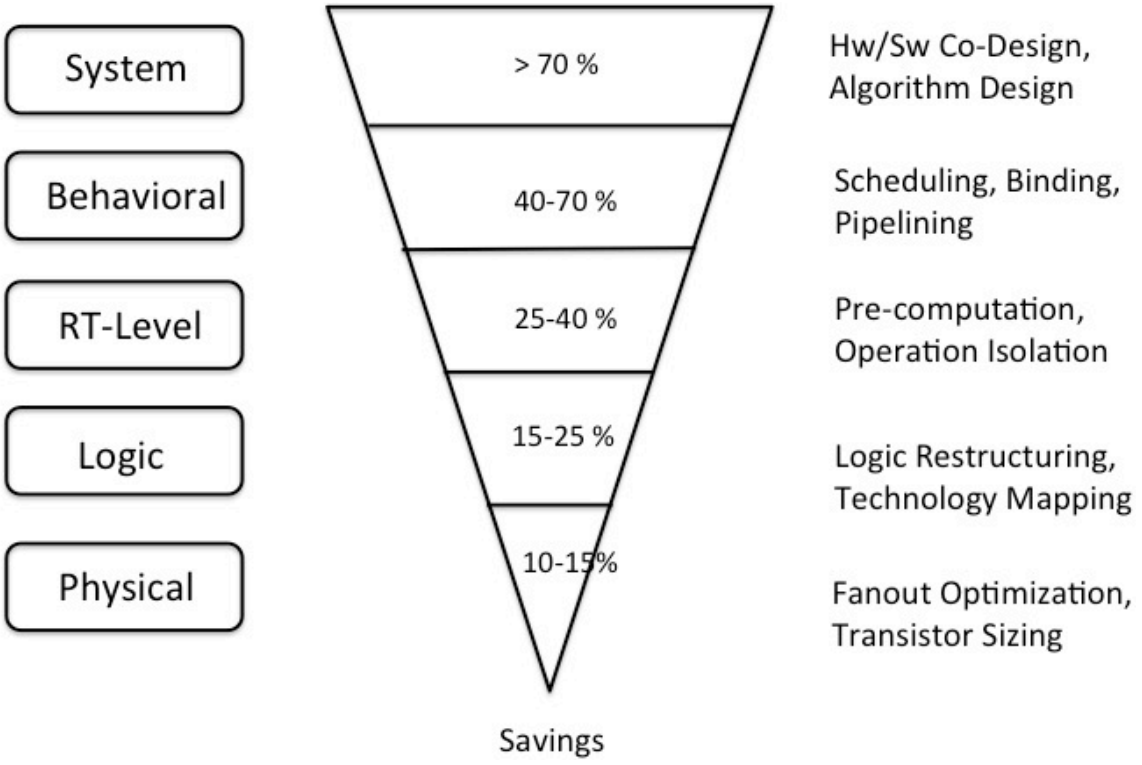

FIGURE 1.3: Power Savings Possible at each stage in Top down VLSI Design Flow

1.3 Organization of the Thesis

The thesis is organized as follows. In Chapter 2, single-mode optical fiber is introduced. Different linear and non-linear phenomena that cause dispersion and its effects on transmitted signal is explained. An end-to-end introduction of a typical CO-OFDM system in case of single and dual-polarization transceiver is done. Commonly used algorithms in CO-OFDM systems are given. Then, need for Multi-Band approach is explained based

on present day DAC/ADC bandwidth and precision available. Literature survey of CO-OFDM offline and online experiments are detailed. Complexity of algorithms (Number of operations) used in both transmitter and receiver is calculated. This gives the state-of-the-art complexity of CO-OFDM systems and provides motivation for reducing complexity from the top, which means starting from low-complexity algorithms to scalable parallel architectures and fixed-point exploration for reduction in resources required.

In Chapter 3, low-complexity algorithms for coarse time synchronization in a dispersive channel is explored. A novel hierarchical low-complexity synchronization algorithm is proposed which provides low Mean Square Error (MSE) performance similar to cross-correlation algorithms. Complexity of the algorithm is compared with previously proposed algorithms. A novel parallel scalable architecture is proposed for coarse time synchronization, which provides high throughput. Proposed real-time architecture is ideal for CO-OFDM system that can receive multiple sample input per cycle and need to match the input rate. Complexity analysis of the proposed architecture is performed and compared with previously proposed real-time architectures for CO-OFDM systems.

In Chapter 4, design of MB-CO-OFDM system given a target rate is shown considering dispersion parameters of the optical channel. An End-to-end parallel transceiver architecture is proposed which can generate and process multiple samples per cycle and easily scale to higher parallel inputs/outputs. Parallel architecture of each block is detailed and savings in resources at the architectural level are shown due to usage of efficient architecture. The architecture exploration is done using CatapultC [21] which is a High Level Synthesis (HLS) tool which accepts input in C and outputs Verilog/VHDL. Fixed-point analysis of the signal processing chain is done which helps in reduction of area for achieving a particular value of bit error rate (BER) at a particular value of Optical Signal-to-Noise Ratio (OSNR). Resources consumed on Xilinx FPGA are reported and break-up of resources consumed for each block is given and compared with previous architectures in terms of scalability and performance.

In Chapter 5, CO-OFDM experiments performed using Arbitrary Waveform Generator (AWG) as transmitter and Digital Storage Oscilloscope (DSO) as receiver are explained. The transition from electrical back-to-back experiment (B2B) to Optical B2B arrangement experiment with optical fiber is explored. BER curves are given for each configuration as a function of SNR. Performance characterization is then done with different LASER used for both transmitter and receiver. Matlab is used for generating and decoding data. Experimental curves of BER are compared with theoretical BER curves for QPSK to validate the setup. Performance of the algorithm and architecture of proposed real-time synchronization algorithm is then explored in the real-time FPGA platform developed as part of 100GFLEX FUI project. The proposed architecture is integrated into this system and performance analysis done with synchronous and asynchronous sampling configurations.

Chapter 6 concludes the thesis by outlining the major contributions done with respect to reduction of computational complexity. Proposals to reach speeds higher than 100 Gb/s

are outlined. Finally perspectives on future work are given.

Chapter 2

CO-OFDM Transceiver System

2.1 Introduction

An introduction to the different blocks of the CO-OFDM transceiver is presented in this Chapter. A CO-OFDM system combines coherent detection and orthogonal multi-carrier modulation to reach higher data rates greater than which is possible by IM-DD systems. CO-OFDM system combines the use of coherent detection, OFDM multi-carrier modulation and electronic dispersion compensation to extract more data rate out of optical fiber channel. Since, the context is optical, the characteristics of single-mode optical fiber are detailed in Section 2.2. Major linear and non-linear phenomena in the optical channel which impair high-speed transmission are explained. Dispersion values of different types of single-mode optical fiber used in core, metro, and access networks are given. Section 2.3 gives the major differences between Wireless and CO-OFDM systems, which helps in understanding the unique challenges posed by the optical fiber and analogue/optical front-ends of the system.

Section 2.4 explains a complete end-to-end CO-OFDM system, giving details separately about digital, analogue/RF, and digital blocks in the subsections that follow. CO-OFDM system is an expensive system compared to IM-DD systems, in terms of optical/analogue/digital components required for its realization. Section 2.5 calculates the resource increase for CO-OFDM system in optical/analogue/digital domains, with detailed analysis done on DSP algorithms used and their complexities. A survey of offline and real-time CO-OFDM experiments is done and then the algorithmic/architectural complexity of those systems is calculated. Section 2.6 lists the observations done by this survey and Section 2.7 concludes the chapter.

2.2 Single-Mode Optical Fiber (SMF)

Single-mode optical fiber is an optical fiber which is designed to carry a single ray of light (mode). The mode defines how the light wave is distributed in space. A typical SMF has core diameter between 8 and 10.5 μm and cladding diameter of 125 μm. SMF allows

a single mode to propagate and is better at retaining the fidelity of light pulse over longer distances. It has lower attenuation and much higher bandwidth than multi-mode optical fibers (MMF). When light pulse travels in SMF, it undergoes pulse width broadening and attenuation along the fiber. Present day attenuation values of SMF fibers are 0.2 dB/km, which requires optical amplifiers (EDFA) only at distances of 50 km apart from each other. EDFA is the most deployed optical amplifier because its amplification window coincides with the band of lowest attenuation (C-band and L-band) in SMF. Different transmission windows used in SMF are listed in Table 2.1. Phenomena which contribute to degradation of signal as it travels through SMF can be grouped into linear and non-linear phenomena. Description about these impairments are given in the following subsections.

TABLE 2.1: DWDM Band Wavelength Range

Band Name	Wavelengths (in nm)
O-Band	1260 - 1360
E-Band	1360 - 1460
S-Band	1460 - 1530
C-Band	1530 - 1565
L-Band	1565 - 1625
U-Band	1625 - 1675

2.2.1 Linear Impairments

Major linear impairments for signal traversing through SMF are fiber attenuation, chromatic dispersion (CD) and polarization mode dispersion (PMD). Each of the phenomenon is explained below.

- Fiber Attenuation: Signal travelling the optical fiber experiences constant attenuation (α_{dB}) as a function of length (L_F), given by

$$\alpha_{dB} = \frac{10}{L_F} log_{10} \frac{P_0}{P} \tag{2.1}$$

  where P_0 is the injected power, P is the received power, and L_F is the length of optical fiber. The attenuation can be classified into intrinsic and extrinsic losses. The intrinsic loss mechanisms are:

  1. Rayleigh scattering - caused by density fluctuations within a fiber.
  2. OH^- Absorption Loss - OH^- is the major impurity responsible for this. It causes attenuation peaks at 1380 nm, 1250 nm, and 950 nm. The peak in the transmission window at 1380 nm is shown in Figure 2.1, which increases the losses to nearly 0.4 dB/km. This peak has been removed by improved manufacturing techniques.

3. Silica Absorption Loss - Pure silica causes absorption loss in two regions above 2000 nm.

The extrinsic loss mechanisms are due to bending loss and connection between two fiber pieces. Figure 2.1 [reproduced from [2]] shows the variation of Fiber attenuation as a function of wavelength. It shows a region of low attenuation in the C-Band and L-Band.

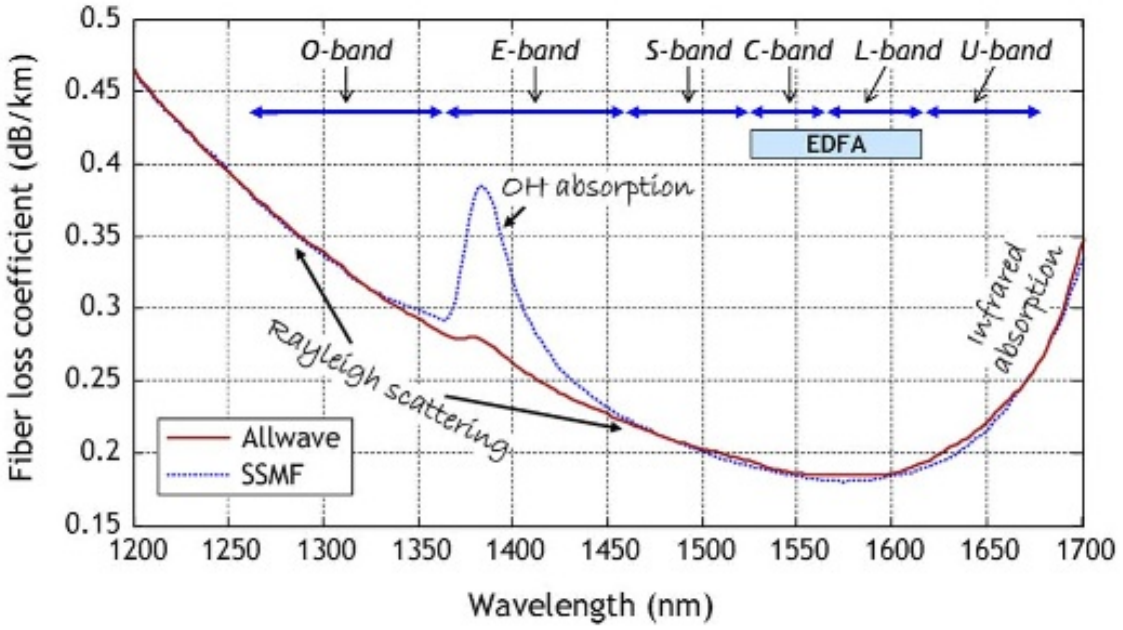

FIGURE 2.1: Fiber loss coefficient vs. different wavelengths for a typical low-loss optical fiber (SSMF) and fiber without the water absorption peak (Allwave). [Reproduced from Essiambre et al.[2]]

- Chromatic Dispersion (CD): Different frequency components of the optical pulse travel with different velocities inside the optical fiber. This leads to pulse broadening and causes interference among neighbouring symbols leading to ISI. It is also called intramodal dispersion. The chromatic dispersion can be expressed as a sum of two components

$$\begin{aligned} CD &= \frac{d\left(-\frac{\lambda^2}{2\pi c}\frac{d\beta}{d\lambda}\right)}{d\lambda} \\ &= -\frac{1}{2\pi c}\left(2\lambda\frac{d\beta}{d\lambda} + \lambda^2\frac{d^2\beta}{d\lambda^2}\right) \\ &= D_M + D_W \end{aligned} \tag{2.2}$$

where D_M - material dispersion, D_W - waveguide dispersion, λ - wavelength of optical signal, β - propagation constant, c - speed of light in vacuum. The units of

CD is $ps/(nm-km)$. The material dispersion is due to refractive index variation in fiber core material which makes different wavelength components travel with unequal speeds. The waveguide dispersion is due to β (propagation constant) being a function of fiber parameters and also wavelength. CD is the major limiting factor for achieving higher single-band data rates using IM-DD systems. As pulse width becomes smaller, ISI increases and complexity of time-domain equalizer in the DD receiver becomes very high.

- Polariztion Mode Dispersion (PMD): The State of Polarization (SoP) of the electric field changes as the signal traverses through the optical fiber. The changes in SOP is random because of fluctuating birefringence. Geometric birefringence and anisotropic stress are the major sources of variation of birefringence. Variation in birefringence means variation of refractive index, which leads to variation in propagation constant (β). PMD is statistical in nature and is given by the following equation:

$$D_p = \frac{\langle (\Delta T)^2 \rangle^{0.5}}{\sqrt{L_F}} \tag{2.3}$$

where D_p - PMD $[\frac{ps}{\sqrt{km}}]$, ΔT - mean square Differential Group Delay (DGD) value, which is a Maxwellian distributed random variable, L_F - length of optical fiber. In case of IM-DD systems, dual polarization is not used. But for systems using dual polarization, it changes channel coefficients and equalizer coefficients have to be updated regularly to accommodate this.

Different SMF types are used based on distances involved in transmission. In undersea network (submarine), distance involved is more than 2000 km. Terrestrial communication networks is divided into core, metro and access networks. Core network covers distances upto few hundreds to thousands of kilometres connecting cities or countries. Metro network connects core and access network, covering several tens of kilometres. Access network provides connectivity to the end users. Typical values of fiber used in all these types of networks with values for fiber attenuation, CD and PMD are given in Table 2.2.

TABLE 2.2: Specifications of commercially available single mode fibers (Corning Fibers)

Fiber Name	ITU-T Naming	CD @1550 nm $(ps/nm-km)$	PMD $ps/\sqrt{km}$	α_{dB} @1550 nm dB/km	Network Usage
PSCF	G.654	20.2	≤ 0.05	0.158	Submarine
SSMF	G.652.D	18	0.1	0.21	Backbone
LEAF	G.655	4.4	≤ 0.04	0.19	Metro
SMF-28	G.652	18	≤ 0.04	0.18	Access

2.2.2 Non-Linear Impairments

If launched power into the optical fiber exceeds several milliWatts in single channel system, then non-linear behaviour of optical fiber becomes significant [22]. In modern WDM technology, high-power semiconductor LASERs and optical amplifiers are used which can exceed several milliWatts. Fiber non-linearities can be classified into two major groups.

1. Kerr Nonlinearities caused by the dependence on the index of refraction on light intensity. It causes pulse distortion due to power variation.
    - Self-phase Modulation (SPM) - Changes in refractive index caused by power variation within the channel leading to pulse distortion.
    - Cross-phase Modulation (XPM) - Pulse distortion caused by variations of power of other wavelength channels in addition to its own channel.
    - Four-wave Mixing (FWM) - New channels are created due to interaction of several wavelength channels. FWM effect depends on chromatic dispersion and powers of interacting channels.
2. Simulated Scattering caused by parametric interaction materials of the fiber and optical light.
    - Simulated Raman Scattering (SRS) - In this scattering, interaction occurs between light and material through vibrations leading to energy transfer from short wavelength channels to long wavelength channels. This causes to crosstalk between channels.
    - Stimulated Brillouin Scattering (SBS) - Interaction occurs between light and material through acoustic waves, leading to coupling with backward propagating waves, which limits the available power per channel.

Non-linear impairments are directly proportional to transmission length (L_F) and inversely proportional to cross-sectional area of the optical fiber. Since non-linear impairments are caused due to higher power signals, the non-linear effects are reduced for attenuated signal. For longer fiber lengths and smaller cross-sectional areas, non-linear interaction is stronger.

2.3 Differences between Wireless-OFDM and CO-OFDM Systems

Differences between wireless and optical OFDM systems is tabulated to contrast the kind of algorithms, architectures which will be specifically required for CO-OFDM systems and which algorithms can be borrowed from Wireless OFDM systems. The main differences are due to channel used, optical/analogue front-end, and data rates involved.

- Wireless channel can have deep spectral nulls in the bandwidth depending on external environment of operation, resulting in frequency selective fading of the signal. The CO-OFDM system uses optical fiber channel which has no spectral nulls in the region of operation.
- Wireless channel varies much faster compared to optical channel whose time constants of variation are of the order of ms. Optical channel is an engineered channel with variations in channel parameters caused by temperature, fiber bending, etc.
- Wireless OFDM systems converts signal from RF to baseband signal using RF-to-analog down converter, while CO-OFDM system converts from optical to RF and then RF to baseband signal using LASER as local oscillator. Because of linewidth of LASER, it results in integer carrier frequency offset (CFO) and rapid phase variations. Rapid phase variations limit the length of symbol size which can be used for OFDM when using digital common phase error (CPE) estimation technique.
- Due to large bandwidth involved for CO-OFDM systems, the data converters (DAC and ADC) become the bottleneck of the system since effective number of bits (ENOB) available at such high bandwidth is also constrained. This imposes resolution constraints on data transmission at DAC and on reception at ADC.
- Data rates of Wireless OFDM systems are in the range of Mb/s, while CO-OFDM systems must support data rates of the order of Gb/s. This difference in data rate makes it necessary for each block to support multiple parallel input/output.

Based on observation of differences between Wireless-OFDM and CO-OFDM systems, the following points are important for the choice of algorithms/architectures for realization of CO-OFDM systems:

- Due to the absence of spectral nulls and channel variations, channel estimation algorithm can be simplified and update rate of the coefficients can be reduced.
- Integer CFO estimation block is mandatory compared to Wireless OFDM systems because of large variation of LASER frequency and efficient phase estimation techniques are required. Continuous monitoring of both frequency and phase is required.
- Due to high data rates involved, highly parallel and scalable architecture are required for all the blocks in the transceiver processing chain. Also, it is necessary to avoid long feedback loops with large delay in critical path of computation.

2.4 Typical CO-OFDM System

CO-OFDM system replaces direct detection in intensity modulation systems with Coherent Detection and single-carrier modulation with OFDM modulation scheme. The use of

coherent detection supports dual polarization in the optical fiber, thus essentially doubling the data rate of a single polarization CO-OFDM system. A brief introduction of Coherent Detection and OFDM modulation is given in subsections below. Next, all the blocks (digital, analogue and optical) of the single polarization and dual polarization CO-OFDM transmitter and receiver are explained.

2.4.1 Coherent Detection

Design of optical transmission system is done by budgeting for different effects in the optical signal processing chain that degrade the signal. The metric used to measure goodness of transmission is done at the receiver by calculation of SNR. SNR can be improved by improvement in noise tolerance at the receiver i.e. better receiver sensitivity. Receiver sensitivity can be defined by the minimum received optical power to keep SNR at the specified level. Receiver sensitivity of CoD and DD are contrasted below.

CoD uses a local oscillator at the receiver which beats at the same frequency as the one at the transmitter to down convert the incoming optical signal. Figure 2.2 [3] shows the noise resilience of CoD and DD methods with different types of symbol mapping. CoD BPSK provides a 4.3 dB improvement in amplified spontaneous noise (ASE) noise tolerance over DD scheme at 40 GBaud signalling rate. Similar tolerance is obtained for dual polarization QPSK at 10 GBaud. This makes CoD a better candidate to work at higher speeds.

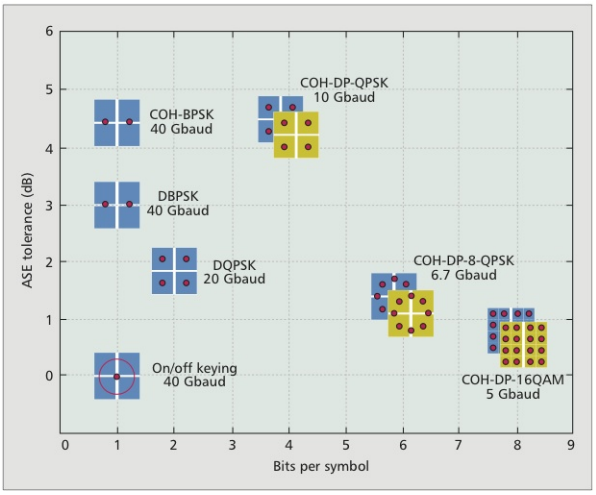

FIGURE 2.2: Tolerance of various phase-amplitude constellations to ASE. Reproduced from [3].

CoD was researched heavily in the 1980s in the quest for providing improved receiver sensitivity by detecting low signal powers caused by fiber loss. The invention of EDFA resulted in low cost optical amplifiers that compensate for fiber loss. Due to its low cost, IM-DD systems gained importance and CoD scheme was neglected. But in optical communication systems operating in excess of 20Gb/s data rates, CD and PMD effects became

very computationally complex to compensate for in DD scheme. CoD started gaining importance because of its ability to give access to optical electrical field. While DD scheme only detects incoming intensity of the optical signal, CoD scheme detects both amplitude and phase of optical signal. This enables the use of Electronic Dispersion Compensation (EDC), which uses DSP techniques for estimation and compensation of these linear dispersion effects of the channel. With the use of DSP, cost of the system can be brought down and flexibility of the system increases significantly. CoD scheme enables higher QAM mapping schemes like QPSK, 16-QAM which increase the bits per symbol. It enables dual polarization schemes which doubles the data rate per band and requires MIMO processing at the receiver to separate the two polarizations. So, with optical community significantly adopting DSP based solutions for very high data rate systems, CoD scheme has seen a revival recently.

2.4.2 OFDM System

OFDM is a special class of multi-carrier modulation system, in which all sub-carriers are orthogonal to each other. The modulation is realized using IDFT at the transmitter and DFT at the receiver. The complexity of implementation of transmitter and receiver is reduced by the usage of FFT.

The major advantages offered by OFDM communication system is as follows:

- Cyclic Prefix (CP) - A portion of end of OFDM symbol is pre-appended to the symbol. This is called CP, whose length is more than maximum delay of the multi-path signal. Hence, ISI effect due to Chromatic Dispersion is eliminated.
- Resistant to FSF - By division of bandwidth into narrow band flat fading channels, it is more resistant to FSF effects of the channel. Frequency nulls can be avoided or bit-power loading can be employed.
- Spectral Efficiency - Efficient usage of bandwidth by spectrum overlap by using orthogonal sub-carriers.
- One-tap Equalization - Due to addition of CP, linear convolution with the channel is converted to circular convolution and hence a single-tap equalizer per sub-carrier is sufficient.

The disadvantages are:

- PAPR - Due to summation of N sub-carriers at the transmitter by IFFT, dynamic range of peak value to mean value varies by a large value [23]. This causes problems for the blocks in the transmission chain to handle such large dynamic range. Clipping at the DAC and introduction of non-linearities at the RF amplifier are some of the effects due to PAPR.

- Sensitive to timing offset - Loss of timing synchronization causes ISI and ICI. Without timing synchronization, other offsets cannot be efficiently estimated and compensated. Symbol synchronization and frame synchronization are essentially the same in case of OFDM.
- Sensitive to Frequency and Phase Offsets - CFO at the receiver causes loss of orthogonality and causes symbol rotation. Phase offset [24] [25] causes rotation of constellation.

Figure 2.3 shows a single band of single polarization/dual polarization CO-OFDM system. Each of the four blocks are explained in the subsections below.

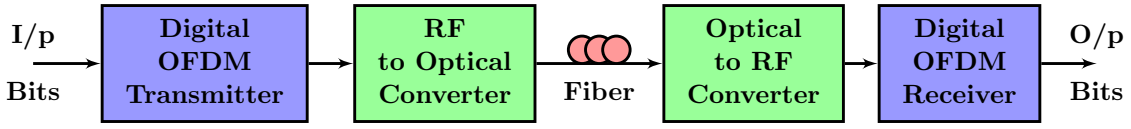

FIGURE 2.3: Single band of a single/dual polarization CO-OFDM system

2.4.3 Digital Transmitter

Figure 2.4 shows the internal blocks of the single polarization digital OFDM transmitter block. In case of dual polarization, the digital transmitter block is repeated used for both polarizations.

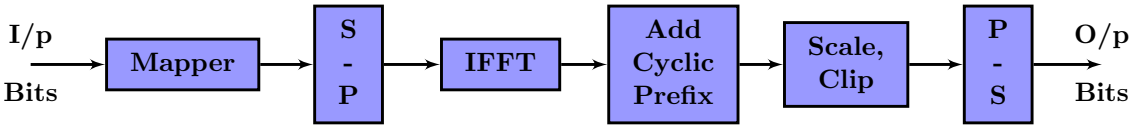

FIGURE 2.4: Digital OFDM Transmitter, S/P - Serial-to-Parallel, P/S - Parallel-to-Serial

- Mapper - It maps bits to symbols. Typical mapping schemes range from BPSK, QAM, 16-QAM, 64-QAM. It is followed by a serial to parallel converter block before the IFFT.
- IFFT - It modulates complex data from frequency domain to time domain. It is the most complex block in the transmitter chain.

$$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X[k] e^{-j2\pi kn/N} \tag{2.4}$$

where $x[n]$ is the time-domain signal, $X[k]$ is the frequency domain signal and N is the size of IFFT.

- Add CP - It adds portion of last part of OFDM symbol to the front. It avoids ISI due to multipath channel when the length of CP is greater than maximum dispersion delay of the channel. It provides immunity against CD of optical fiber.
- Scale, Clip - Output of IFFT is scaled and clipped to fit in the input voltage range of the DAC. Clipping value must be chosen to minimize clipping distortion as well as quantization noise.

2.4.4 RF-to-Optical Up Converter

Figure 2.5 shows the RF-to-Optical Converter for a single polarization CO-OFDM system, while Figure 2.6 corresponds to dual polarization CO-OFDM system.

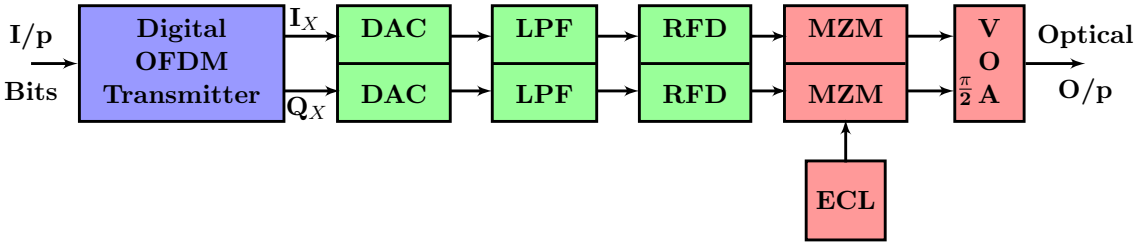

FIGURE 2.5: Single Polarization RF-to-Optical Up Converter. $\mathbf{I}_X$ - Real Part of X-Polarization, $\mathbf{Q}_X$ - Imaginary Part of X-Polarization, **DAC** - Digital-to-Analog Converter, **LPF** - Low Pass Filter, **RFD** - RF Driver, **MZM** - Mach-Zender Modulator, **ECL** - External Cavity LASER, **VOA** - Variable Optical Amplifier.

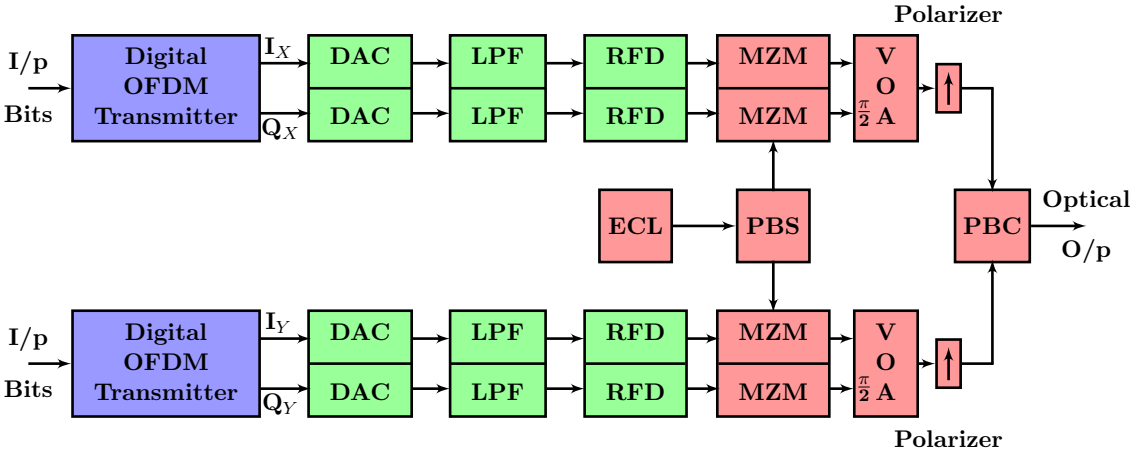

FIGURE 2.6: Dual Polarization RF-to-Optical Up Converter. $\mathbf{I}_X$ - Real Part of X-Polarization, $\mathbf{Q}_X$ - Imaginary Part of X-Polarization, $\mathbf{I}_Y$ - Real Part of Y-Polarization, $\mathbf{Q}_Y$ - Imaginary Part of Y-Polarization, **DAC** - Digital-to-Analog Converter, **LPF** - Low Pass Filter, **RFD** - RF Driver, **MZM** - Mach-Zender Modulator, **ECL** - External Cavity LASER, **PBS** - Polarization Beam Splitter, **VOA** - Variable Optical Amplifier, **PBC** - Polarization Beam Combiner.

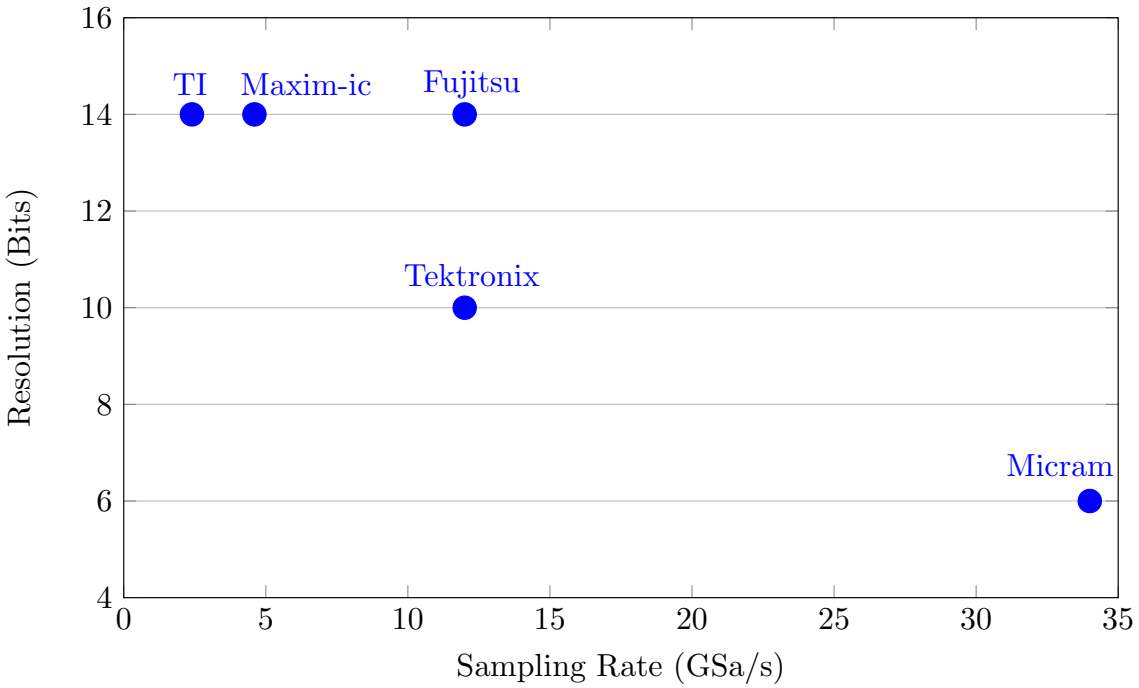

FIGURE 2.7: Resolution vs. Sampling Rate for fastest DAC available. GSa/s - Giga Samples/second.

- DAC - DAC converts digital output of IFFT to analogue output. Present day DACs bandwidth and resolution lag behind the requirements for 100 Gb/s single-band CO-OFDM system. A survey of the fastest DAC available in the market is shown in Figure 2.7. Fastest DAC available has a sampling rate of around 34 GSamples/s with a resolution of 6 bits. To reduce the constraints on DAC/ADCs, multi-band CO-OFDM has been proposed to achieve a total data rate of 100 Gb/s in case of 100 Gb Ethernet.
- Low Pass Filter (LPF) - It filters the output signal with a cut-off frequency near the Nyquist frequency of the DAC sampling frequency.
- RF driver - This amplifies the electrical signal after low pass filtering and output modulates optical carrier in MZ Modulator.
- MZM - The carrier frequency supplied by External Cavity LASER (ECL) module is modulated by the I/Q electrical signal.
- Variable Optical Amplifier (VOA) - The real and imaginary signals are combined and amplified by the optical amplifier. The output signal is fed to Polarizer in case of dual-polarization system.
- Polarization Beam Combiner (PBC) - The modulated signal amplitude is controlled by VOA and the two polarizations are combined and input to single mode optical fiber channel in case of dual polarization CO-OFDM system.

2.4.5 Optical-to-RF Down Converter

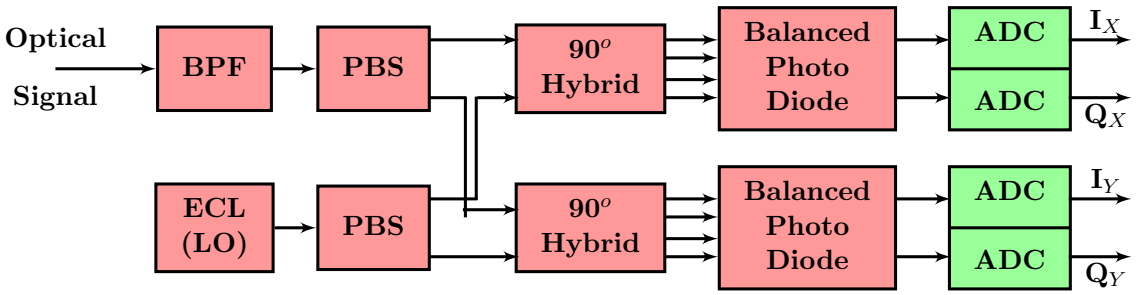

FIGURE 2.8: Optical-to-RF Down Converter. **BPF** - Band Pass Filter, **ECL** - External Cavity LASER, **LO** - Local Oscillator, **PBS** - Polarization Beam Splitter, **ADC** - Analog-to-Digital Converter, $\mathbf{I}_X$ - Real Part of X-Polarization, $\mathbf{Q}_X$ - Imaginary Part of X-Polarization, $\mathbf{I}_Y$ - Real Part of Y-Polarization, $\mathbf{Q}_Y$ - Imaginary Part of Y-Polarization.

Figure 2.8 shows the front end of the optical receiver for receiving either single/dual polarized optical signal. It shows direct down conversion architecture, where conversion from optical to analogue is direct without any intermediate RF frequency. The Band Pass Filter (BPF) selects the band for processing. The filtered signal is down converted by using LASER frequency which is tuned to center frequency of the band. The optical signal's amplitude and phase information is detected by balanced photodiode circuit and then sampled by ADC and converted to digital domain. The bandwidth of ADC is the limiting factor. Oversampling by a large factor is not possible due to this limitation. Generally, for CO-OFDM systems, an oversampling factor of 1.2 is used. A survey is done of the fastest available ADC in the market as shown in Figure 2.9. The fastest available ADC has sampling rate of 56 Gb/s with resolution of 8 bits.

2.4.6 Digital OFDM Receiver

Figure 2.10 shows the digital part of the receiver. The major components are:

- Coarse Time Synchronization - It detects start of OFDM frame by detecting start of training symbol. Estimation of fractional carrier frequency offset (CFO) is done.
- Fractional Frequency Synchronization - Using the fractional CFO is estimated. CFO compensation is done by multiplying input signal with estimated CFO. It receives integer CFO from integer CFO estimation block which is present after FFT block.
- Remove CP - After CFO compensation, cyclic prefix (CP) is removed and N samples are fed into FFT block.

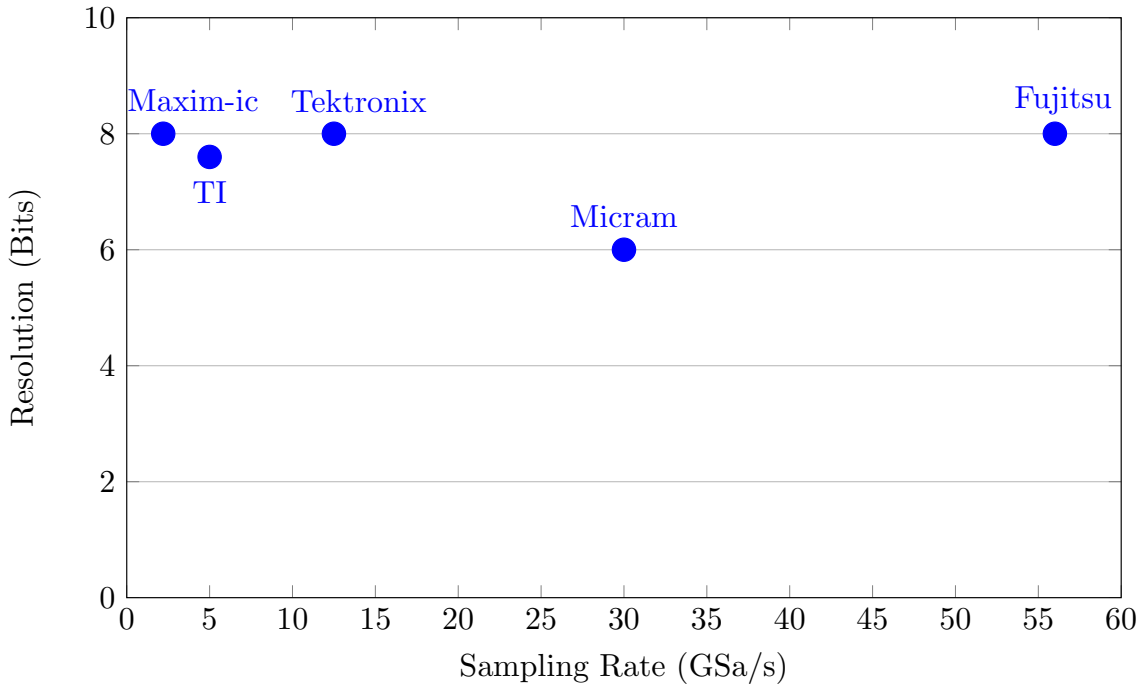

FIGURE 2.9: Resolution vs. Sampling Rate for fastest ADC available. GSa/s - Giga Samples/second.

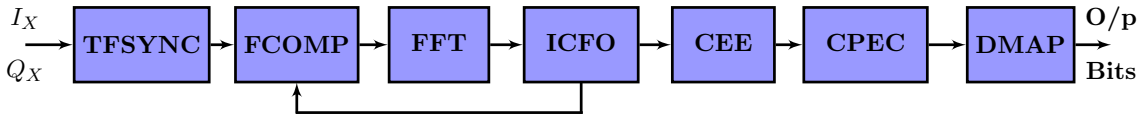

FIGURE 2.10: Digital Receiver of PDM-CO-OFDM System. **TFSYNC** - Time Frequency Synchronization, **CFO** - Carrier Frequency Offset, **FCOMP** - CFO Compensation, **FFT** - Fast Fourier Transform, **ICFO** - Integer CFO Estimation, **CEE** - Channel Estimation & Equalization, **CPE** - Common Phase Error, **CPEC** - CPE Estimation & Compensation, **DMAP** - Demapper.

- FFT - It converts input time domain samples to frequency domain output samples. It is the most complex block in the receiver chain.

$$X[k] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x[n] e^{j2\pi kn/N} \tag{2.5}$$

- Integer Frequency Estimation - In case of CO-OFDM systems, Integer CFO is present due to variations of LASER frequency by large amount. Integer CFO is estimated by use of training symbol. The estimated value is fed back to CFO compensator block.

- Channel Estimation & Equalization - Channel Estimation is done using training symbols and then tracking is done either using decision-directed equalizers like LMS Equalizers or averaging techniques in time/ frequency domain.

- Common Phase Error Estimation - Phase Error in the OFDM symbol caused due to LASER's rapid variations phase is estimated using pilot symbols dedicated in every OFDM symbol. Compensation is done by multiplication by exponential multiplication.
- Demapper - It converts received complex data to symbols of used constellation.

Since the data converters (DAC, ADC) form the bottleneck with respect to sampling frequency and also resolution, multi-band CO-OFDM (MB-CO-OFDM) is proposed to reduce the pressure on the signal converters. Also, MB-CO-OFDM helps in the realization of architectures on FPGA since the maximum frequency attained on an FPGA is order of magnitude lesser than that of DAC/ADC. MB-CO-OFDM divides the total optical bandwidth into smaller electrical bandwidths which can be handled by DAC/ADC and target rate of 100 Gb/s is attained by the use of multiple bands working in parallel. In this thesis, all the designs are for single-band single-polarization CO-OFDM block. Dual-Polarization is indicated when it is applicable. Then, the total target data rate of 100 Gb/s is achieved by using dual polarization multiple bands.

2.5 Complexity Analysis of the System

CoD in CO-OFDM increases the cost of the system compared to IM-DD system by increasing the number of optical/analogue components required. Table 2.3 gives the number of components for transceiver for CO-OFDM, CO-QPSK and IM-DD system. From the table, it can be seen that there is a significant increase of resources for CoD systems and where this increase compensates for bandwidth increase, it is beneficial to use. Hence, long-haul core and submarine networks are better candidates at present for adoption of CoD detections schemes compared to metro and access networks.

Table 2.3: Cost of Optical Transceiver for CO-OFDM, CO-QPSK and IM-DD Systems

System	LASER	Optical Modulator	Photo-Diode	DAC/ ADC	PBS PBC
DP-CO-OFDM	4	4	4	4/4	1/1
SP-CO-OFDM	2	2	2	2/2	1/1
DP-CO-QPSK	4	4	4	0/4	1/1
SP-CO-QPSK	2	2	2	0/2	1/1
IM-DD	1	1	1	0/1	0/0

For finding the increase in complexity in the digital part, a full complexity analysis is done. The analysis is done at two levels. First, the algorithmic complexity of algorithms

used in transmitter and receiver is calculated. Algorithmic complexity gives the total number of real multiplications and additions required for computation of single sample of output. For example, total number of complex multiplications and additions required for one output of IFFT is expressed in terms of size of IFFT (N). Then, the architectural complexity of the algorithms is calculated for throughput of one output every clock cycle. Throughput of one output clock is necessary to support high data rates and to avoid large buffer memory. Architectural complexity involves calculation of number of real multipliers and adders required for realization of the algorithm.

2.5.1 Digital Transmitter

The major computational block in the digital transmitter is IFFT. The algorithmic complexity of radix-$2/4/2^2$ or split radix IFFT block is given in Table 2.4, which gives total number of multiplications and additions as a function of IFFT size (N). Architectural complexity of Feedforward pipelined architectures using radix-$2/4/8$ in terms of total number of multipliers, adders and memory requirement is given in Table 2.5.

TABLE 2.4: Algorithmic Complexity in terms of size of IFFT/FFT N.

Radix	Real Multiplications	Real Additions
Radix-2	$2N \cdot \log_2 N$	$3N \cdot \log_2 N$
Radix-4	$\frac{3}{2}N \cdot \log_2 N$	$\frac{5}{4}N \cdot \log_2 N$
Radix-2^2	$\frac{3}{2}N \cdot \log_2 N$	$\frac{5}{4}N \cdot \log_2 N$
Split-Radix	$\frac{4}{3}N \cdot \log_2 N$	$\frac{8}{3}N \cdot \log_2 N$

Survey of previously reported real-time transmitter with offline receiver is done. The objective is to calculate transmitter's architectural complexity, which inherently comes down to calculation of IFFT complexity. Table 2.6 lists real-time CO-OFDM experiments on standard single mode fiber (SSMF) which have achieved gigabit per second using real-time implementation on an FPGA.

The computational complexity of the proposed real-time solutions is given in Table 2.7. Proposal by Inan et al. [31] uses radix-2 IFFT for larger parallel factor of 64, which is inefficient considering higher radix can be used at such high parallel output. Proposal by Schmogrow et al. [29] does not use multipliers, but huge number of adders and LUTs. Since, all multiplier combinations are stored in memory, it is limited to small size (N) of IFFT of 64. This approach is not scalable to higher speeds.

TABLE 2.5: Architectural Complexity of feedforward pipelined IFFT/FFT for 2/4/8-Parallel Outputs as a function of IFFT/FFT size (N). MDC - Multipath Delay Commutator.

Radix	Architecture Type	Real Multipliers	Real Adders	Total Memory
2-PARALLEL INPUT/OUTPUT				
Radix-2 [26]	MDC	$8(\log_4 N - 1)$	$12\log_4 N - 4$	$2N$
4-PARALLEL INPUT/OUTPUT				
Radix-2	MDC	$16(\log_4 N - 1)$	$24\log_4 N - 8$	$2N$
Radix-4 [27]	MDC	$12(\log_4 N - 1)$	$22\log_4 N - 6$	$2N$
8-PARALLEL INPUT/OUTPUT				
Radix-2 [28]	MDC	$32(\log_4 N - 1)$	$48\log_4 N - 16$	$2N$
Radix-4	MDC	$24(\log_4 N - 1)$	$44\log_4 N - 12$	$2N$
Radix-8 [27]	MDC	$24(\log_4 N - 7)$	$44\log_4 N - 14$	$\frac{32N}{7}$

TABLE 2.6: Real-time CO-OFDM Transmitter Implementation

Reference	Bandwidth (GHz)	Data Rate (Gb/s)	IFFT size (N)	Cyclic Prefix	Year
Schmogrow et al. [29]	25.4	101.5	64	0	2011
Inan et al. [30]	11.9	23.9	1024	64	2011
Inan et el. [31]	23.4	93.8	1024	64	2011

TABLE 2.7: Computational Complexity for CO-OFDM Transmitter

Reference	Real Multipliers	Real Adders	IFFT size (N)	Radix used
Schmogrow et al. [29]	0 (uses LUTs to store values)	3776	64	Radix-2/4
Inan et al. [31]	844	1732	1024	Radix-2 (64-Parallel)

2.5.2 Digital Receiver

For all blocks in the receiver, algorithmic and architectural complexity is calculated for single/multiple parallel output. The major blocks of the receiver are: Time/Frequency Synchronization, CFO Compensation, FFT, Integer CFO Estimation, Channel Estimation & Equalization, CPE Estimation & Compensation, and Demapper. For architectural complexity comparison, only two papers are available which have implemented real-time

CO-OFDM receiver architecture on FPGA, Kaneda et al. [14] and Chen et al. [32]. All architectural complexity comparisons are done with these two papers wherever it is relevant.

2.5.3 Time/Frequency Synchronization

Algorithmic complexity for some of the algorithms which can be used for coarse time synchronization are tabulated in Table 2.8, which shows number of operations required for calculation of single timing metric point. The algorithms Schmidl-Cox, Minn-Bhargava and Shi-Serpedin have an iterative form of equation for correlation operation, which makes number of operations independent of size of IFFT/FFT (N). The algorithms of Park, Choi and Zhou do not have iterative form for the correlation operation, hence number of operations depends on size of N.

TABLE 2.8: Algorithmic Complexity of Coarse Time Synchronization Algorithms. Calculations count only correlation operation and not the energy calculation.

Algorithm	Real Multiplications	Real Additions
Schmidl-Cox [33]	8	8
Minn-Bhargava($L = 4$) [34]	12	12
Shi [35]	54	57
Park [36]	$2(N + 2)$	$2(N + 1)$
Choi [37]	$2N$	$2(N - 1)$
Zhou [38]	$2N$	$2(N - 1)$

Although the algorithmic complexity of auto-correlation algorithms is better for high-parallel outputs compared to cross-correlation algorithms, it does not translate into hardware savings and scalability in architectural space. Direct parallelization of auto-correlation equation results in huge amount of resources and implementation of iterative equation assumes operating frequency of digital circuit is the same as ADC sampling frequency which is not, in the case of CO-OFDM receiver. In case of CO-OFDM receiver, ADC frequency is of the order of GHz and FPGA operating frequency is around 250-300 MHz. Table 2.9 shows the proposals for parallel coarse time synchronization used in real-time CO-OFDM systems. Only two proposals are available, Kaneda et al. and Chen et al. Kaneda et al. proposed for $R = 16$-parallel output, while Chen et al. proposed for $R = 8$-parallel output. Their proposal is extended for $R = 2, 4, 8, 16$-parallel output to show the trend. The amount of resources shown is required for computation of only auto-correlation part of the timing metric. They do not include energy calculation and division by energy of training symbol. Also, both the proposals do not provide fractional CFO estimation. Table 2.9 indicates that further improvement in scalable parallel architectures for coarse timing synchronization is required.

TABLE 2.9: Architectural Complexity of Coarse Time Synchronization Algorithms

Author	FFT Size (N)	Training Sym. Size (M)	Algorithm Used	Real Multipliers	Real Adders
2-PARALLEL INPUT/OUTPUT					
Kaneda et al.	128	32	Auto-Corr.	8	68
Chen et al.	128	128	Cross-Corr.	0	508
4-PARALLEL INPUT/OUTPUT					
Kaneda et al.	128	32	Auto-Corr.	16	136
Chen et al.	128	128	Cross-Corr.	0	1016
8-PARALLEL INPUT/OUTPUT					
Kaneda et al.	128	32	Auto-Corr.	32	272
Chen et al. [32]	128	128	Cross-Corr.	0	2032
16-PARALLEL INPUT/OUTPUT					
Kaneda et al. [14]	128	32	Auto-Corr.	64	544

Total CFO can be expressed as

$$\widehat{\epsilon}_{total} = \widehat{\epsilon}_{frac} + \widehat{\epsilon}_{int} \tag{2.6}$$

$$\widehat{\epsilon}_{total} = \frac{\widehat{\phi} L}{\pi N} + \frac{2z}{N} \tag{2.7}$$

where $\widehat{\epsilon}_{frac}$ is the fractional CFO, $\widehat{\epsilon}_{int}$ is the integer CFO, L is the number of repeating parts of training symbol, N is the size of IFFT/FFT. Fractional CFO Estimation is done with the help of training symbol used for coarse synchronization, given by

$$\widehat{\epsilon}_{frac} = \frac{2}{\pi} \cdot \underline{/P[\widehat{\eta}_{start}]} \tag{2.8}$$

where $\widehat{\epsilon}_{frac}$ is the fractional CFO estimate $\widehat{\eta}_{start}$ is the estimate of start of OFDM symbol, $P[\eta_{start}]$ is the auto-correlation function at the index of $\widehat{\eta}_{start}$. $\widehat{\epsilon}_{frac}$ gives CFO in terms of sub-carrier spacing ($\frac{B}{N}$), B is the bandwidth of the OFDM signal, N is the size of IFFT/FFT. The range of CFO estimation of Schmidl-Cox algorithm is ± 1, and for Minn-Bhargava algorithm ($L = 4$) is ± 2. Algorithmic complexity of CFO Estimation is arc tangent calculation and architectural complexity is LUT implementation of arc tangent function. No real-time proposals for fractional/integer CFO estimation are proposed in literature.

2.5.4 CFO Compensation

Compensation of CFO is done by exponential multiplication by using the estimated CFO,

$$x_c[n] = x[n] \cdot e^{-j2\pi n(\widehat{\epsilon}_{frac} + \widehat{\epsilon}_{int})/N} \tag{2.9}$$

where $x_c[n]$ is the CFO compensated signal, $x[n]$ is the input signal, $\widehat{\epsilon}_{int}$ is the CFO integer estimated, n is the time index $n \in [0, N-1]$, N is the size of IFFT/FFT. For N samples in a single OFDM symbol, algorithmic complexity is $4N$ real multiplications and $2N$ real additions. Architectural complexity for R-parallel output is $5R$ real multipliers and $3R$ real adders, where R is the number of parallel outputs.

2.5.5 FFT

The FFT converts received data from time domain to frequency domain. It is the most computationally intensive block in the receiver signal processing chain. Algorithmic complexity is shown in Table 2.4 and architectural complexity is shown in Table 2.5. Kaneda et al. used in-built Altera FFT for $N = 128$ size FFT, which uses 24 real multipliers and totally 384 multipliers were used for decoding 16-parallel inputs.

2.5.6 Integer CFO Estimation

Because of large variation in LASER's frequency, integer CFO is introduced. No previous real-time architecture proposals for Integer CFO Estimation in lietrature. There are two methods for estimating integer CFO.

- Method 1 - The method uses either two symbols [33] or one symbol [39]. In case of two symbols/one symbol, the even sub-carriers in the symbols are related by a fixed phase factor. The timing metric for detection of integer CFO is

$$B_1(g) = \frac{\left|\sum_{k\in X} x^*_{1,k+2g} v^*_k x_{2,k+2g}\right|^2}{2\left(\sum_{k\in X} |x_{2,k}|^2|\right)^2} \tag{2.10}$$

  where integer g spans the range of possible frequency offsets, $X = -W, .., 2, ..., W$ is the set of indices for even frequency components, W is the number of even frequencies with the PN sequence. The index corresponding to the maximum value of $B_1(g)$ gives the integer CFO. Algorithmic/Architectural Complexity is given in Table 2.10.

- Method 2 - The method uses cross-correlation with known sequence to estimate integer CFO. The timing metric is given by

$$B_2(g) = \frac{\left|\sum_{k\in X} x_{k+2g} y^*_{k+2g}\right|^2}{\sum_{k\in X} |x_{k+2g}|^2} \tag{2.11}$$

  where y is the known sequence, x is the received sequence, integer g spans the range of possible frequency offsets, $X = -W, .., 2, ..., W$ is the set of indices for even frequency components. The index corresponding to maximum value of $B_2(g)$ gives the integer CFO estimate. Algorithmic/Architectural Complexity is given in Table 2.10.

TABLE 2.10: Algorithmic/Architectural Complexity for integer CFO Estimation. R - number of parallel outputs.

Algorithm	Real Multiplications	Real Additions	Real Multipliers	Real Adders
Method 1	$11NW/2$	$2NW$	$10R$	$7R$
Method 2	$3NW$	$3NW/2$	$6R$	$5R$

2.5.7 Channel Estimation and Equalization

2.5.7.1 Least Squares (LS)

Channel estimation using LS approach is given for both single/dual polarizations.

1. Single polarization - Consider the received signal,

$$R_{km} = H_k \cdot c_{km} + N_{km} \tag{2.12}$$

where R_{km} is the k^{th} sub-carrier of m^{th} received OFDM symbol, H_k is the channel response for k^{th} sub-carrier, c_{km} is the k^{th} sub-carrier of m^{th} transmitted OFDM symbol, N_{km} is the additive noise. Using the training symbol, channel frequency response can be estimated according to LS criterion using optimization criterion [40],

$$\hat{H}_k = \arg_{min} ||R_k - H_k c_k||^2. \tag{2.13}$$

$\hat{H}_k$ can be calculated using

$$\hat{H}_k = \frac{R_k^* c_k}{|R_k|^2} \tag{2.14}$$

where $k \in [0, 1, .., N_{usc}]$, N_{usc} is the used sub-carrier number. Since optical channel does not have frequency nulls in the band of operation, the problem of noise enhancement is avoided. Algorithmic/Architectural Complexity for LS channel estimation is given in Table 2.11. The equalization is done using $\hat{H}_k$ given by

$$C_{km} = R_{km} \cdot \hat{H}_k. \tag{2.15}$$

where C_{km} is the equalized signal.

2. Dual Polarization - In dual polarization system, the received signal can be written as

$$R_{km,x} = H_k^{xx} \cdot c_{km,x} + H_k^{xy} \cdot c_{km,y} \tag{2.16}$$

$$R_{km,y} = H_k^{yx} \cdot c_{km,x} + H_k^{yy} \cdot c_{km,y} \tag{2.17}$$

There are four coefficients to be calculated, which can be simplified by transmitting training symbols in only one polarization at a time. The system of equations using

this scheme can be written as

$$\begin{bmatrix} R_{km,x} & R_{k(m+1),x} \\ R_{km,y} & R_{k(m+1),y} \end{bmatrix} = \begin{bmatrix} H_k^{xx} & H_k^{xy} \\ H_k^{yx} & H_k^{yy} \end{bmatrix} \cdot \begin{bmatrix} c_{km,x} & 0 \\ 0 & c_{k(m+1),y} \end{bmatrix} \tag{2.18}$$

Estimation of the coefficients is given by

$$\hat{H}_k^{xx} = \frac{R_{km,x}}{c_{km.x}}, \quad \hat{H}_k^{xy} = \frac{R_{k(m+1),x}}{c_{k(m+1),y}} \tag{2.19}$$

$$\hat{H}_k^{yx} = \frac{R_{km,y}}{c_{km,x}}, \quad \hat{H}_k^{yy} = \frac{R_{k(m+1),y}}{c_{k(m+1),y}} \tag{2.20}$$

The equalization can be done using $\hat{H}_k$

$$C_{km} = \hat{H}_k^{-1} \cdot R_{km} \tag{2.21}$$

where C_{km} is the equalized signal, $\hat{H}_k^{-1}$ is the inverse of 2x2 $\hat{H}_k$ matrix. Since the $\hat{H}_k$ is a unitary matrix, the inverse calculation can be done by using a Hermitian transpose. Algorithmic/Architectural complexity is given in Table 2.11.

2.5.7.2 Normalized Least Mean Squares (NLMS)

After initial estimation using training symbols by LS estimation, NLMS method can be used to track the channel. The equations used for single polarization are

$$e_k = \hat{R}_k - R_k^{ideal} \tag{2.22}$$

$$|R|_k^2 = k_1 \cdot |R|_{k,old}^2 + (1 - k_1) \cdot |R|_k^2 \tag{2.23}$$

$$\hat{H}_k = \hat{H}_{k,old} + step \cdot e_k \cdot \frac{R_k^*}{|R|_k^2} \tag{2.24}$$

where e_k is the error between received symbol (R_k) and ideal constellation symbol (R_k^{ideal}), k_1 is the coefficient for updating energy, $|R|_{k,old}^2$ is the old value of energy, $step$ is the coefficient for updating equalizer coefficients, $\hat{H}_{k,old}$ is the old value of equalizer coefficient. Algorithmic Complexity for NLMS Estimation is given in Table 2.11. Equalization involves complex multiplications with channel estimated and its algorithmic/architectural complexity is given in Table 2.11. Kaneda et al. simplified the channel estimation significantly by using look-up table implementation. No multipliers were required for channel estimation.

2.5.8 CPE Estimation and Compensation

CPE Estimation [14] is done by comparing and averaging received pilot phases with reference phase to calculate phase noise. Compensation is done by rotating the symbols using

TABLE 2.11: Algorithmic/Architectural Complexity for Channel Estimation and Equalization. R - number of parallel outputs.

Algorithm	Real Multiplications	Real Additions	Real Multipliers	Real Adders
Least -Squares Single Pol.	$8N$	$3N$	7R	3R
Least -Squares Double Pol.	$32N$	$12N$	28R	12R
NLMS Single Pol.	$11N$	$8N$	5R	6R
NLMS Dual Pol.	$22N$	$16N$	10R	12R
Channel Equalization Single Pol.	$4N$	$2N$	8R	4R
Channel Equalization Double Pol.	$8N$	$4N$	16R	8R

the phase error calculated. The method is as follows

$$e = \frac{1}{N_p} \sum_{m=0}^{N_p-1} r^*[m] \cdot c[m] \tag{2.25}$$

$$\phi_{err} = \angle e \tag{2.26}$$

where e is the complex error vector, N_p is the number of pilots in one OFDM symbol, $r[m]$ is the received signal, $c[m]$ is the reference pilot symbol. The CPE compensation is done by multiplying the input signal by $e^{\phi_{err}}$. Algorithmic/Architectural complexity is given in Table 2.12. Kaneda et al. simplified CPE estimation by using LUT implementation, which avoided multipliers.

TABLE 2.12: Algorithmic/Architectural Complexity for CPE Estimation and Compensation. R - number of parallel outputs.

Algorithm	Real Multiplications	Real Additions	Real Multipliers	Real Adders
CPE Estimation	$4N_p$	$2N_p + (N_p - 1)$	4R	4R
CPE Compensation	$4N$	$2N$	4R	2R

2.5.9 Demapper

Demapper - It maps incoming complex symbols to symbols of a constellation. It involves comparisons with reference symbols of the constellation and calculation of distance. In case of QPSK de-mapping, it can be reduced to checking positive/negative sign and mapped to one of QPSK symbol.

2.6 Observations

From the algorithmic and architectural complexity calculations, it can be seen that IFFT/ FFT are the major resource hungry blocks. With the adoption of multi-band CO-OFDM to reach the total target data rate of 100 Gb/s, resource savings obtained from one polarization single-band is multiplied by the total number of sub-bands used. Hence, effort targeted towards resource optimization by low-complexity algorithm/architecture goes a long way in reduction of digital computational complexity of CO-OFDM. From the survey, it has been found that there is no low-complexity parallel architecture proposed for time synchronization. Also, no proposal for efficient integer CFO estimation. Channel Estimation is one more block which occupies significant area and hence needs to be optimized. Hence this thesis directs its efforts towards low-complexity scalable algorithms/architectures for single-band single-polarization CO-OFDM system. The only resource shared from transition from single-polarization to dual-polarization is channel estimation. Other than that, all the blocks are replicated in both polarizations.

2.7 Conclusions

State-of-the-art survey of real-time CO-OFDM systems show that transmitters supporting large rates have been built. But, there also optimization of IFFT architecture and scalability has not been explored. In case of real-time CO-OFDM receiver, only two major publications have appeared which explore the complexity of the system. End-to-end parallel architectures have not been explored and scalability also is an interesting option. Chapter 3 explores timing synchronization from a low-complexity algorithmic and architecture standpoint and Chapter 4 explores end-to-end parallel architectures for the complete CO-OFDM receiver. Chapter 5 details the experiments performed using proposed low-complexity algorithms and performance characterized.

Chapter 3

Timing Synchronization in OFDM Systems

3.1 Introduction

OFDM systems are sensitive to timing, carrier frequency offset (CFO)[41][42] and phase offset [43]. Loss of timing synchronization causes inter-carrier interference (ICI) and inter-symbol interference (ISI). It also leads to reduced accuracy in carrier frequency offset estimation and causes sub-carrier dependent phase rotation after FFT. Uncompensated CFO also causes rotation of sub-carriers proportional to frequency offset. Thus, loss of timing and frequency synchronization reduces the advantages provided by single-carrier OFDM over single-carrier systems.

In Section 3.2, a survey of timing synchronization algorithms proposed for Wireless OFDM systems is done. Survey was done to look at possible improvements possible in timing estimation. This led to a novel proposal of hierarchical low-complexity synchronizer for Wireless OFDM systems which is given in Section 3.3. Performance of the proposal is evaluated in Section 3.4 in an ISI channel. In Section 3.5, the proposal is adapted to optical channel with modifications to reduce complexity. The performance of adapted proposal is evaluated using single mode optical fiber channel in Section 3.6. In Sections 3.9 and 3.10, the proposed streaming parallel architectures for synchronization algorithm is explained. Architectural complexity of the proposed architectures is calculated to show the scalability of the architecture and compared with previous proposals. Section 3.12 concludes the chapter.

3.2 Timing Synchronization in Wireless OFDM Systems

In the OFDM transmitter, input bits are mapped to QAM constellation by mapper block. These mapped symbols are considered to be in frequency domain. The mapped symbols are passed through IFFT and converted to time domain. Every OFDM symbol (training, data) is passed through IFFT and converted to time domain before transmission. Since

timing synchronization operation is done before IFFT in the receiver, it is performed on time domain data. All the methods (auto-correlation/cross-correlation) proposed for timing synchronization work on time domain data to obtain the starting point estimation of OFDM symbol. Some of the methods also give an coarse estimate of carrier frequency offset (CFO) which is used to compensate the received OFDM symbol before passing it through the FFT.

Many methods have been proposed for initial timing synchronization previously. The synchronization methods have either been based on cyclic prefix [44] or known preamble symbols [33][34][35][36][37]. Cyclic prefix (CP) based methods utilize the cyclic prefix length for synchronization which constitutes only a small part of the OFDM symbol. In a multipath channel, cyclic prefix samples are corrupted due to ISI. This leads to reduction in the number of uncorrupted samples for synchronization which leads to reduction in the accuracy of estimation.

Preamble-based synchronization methods utilize specially designed training symbol for achieving synchronization. Preamble symbols are designed with specific structure for maximizing the detection of start point of the OFDM symbol. Schmidl et al. [33] proposed a training symbol (TS) which consists of two repetitive parts. The correlation peak of the timing metric has a plateau equal to the length of cyclic prefix in AWGN channel. The length of plateau is equal to length of the uncorrupted portion of the cyclic prefix in frequency selective (ISI) channels. The training symbol also allows a CFO estimation of $\pm$ 1 subcarrier spacing. Minn et al. [34] proposed an algorithm consisting of multiple repetitive parts with specific sign pattern for timing synchronization. The algorithm has steeper timing roll-off compared to Schmidl's timing metric roll-off. CFO estimation was done using algorithm by Morelli [45]. The symbol has a CFO estimation range of $\pm\frac{L}{2}$ sub-carrier spacings, where L is the number of repetitive parts in the TS. Shi and Serpedin [35] proposed a four part TS which is similar to Minn's, but the algorithm used for timing estimation is more generalized and uses all repetitive parts of the TS. The CFO estimation range is $\pm$ 2 sub-carrier spacing.

Park et al. [36] proposed a new training symbol which utilizes conjugate symmetry property for timing synchronization. The timing metric has a steeper roll-off compared to Minn's method. Choi et al. [37] proposed a training symbol for multipath channels which also utilizes conjugate symmetry for correlation operation. The symbol is based on CAZAC (Zadoff-Chu) sequence, which has constant amplitude and has much better correlation properties than m-sequences. The timing metric has impulse like roll-off at the correct starting point. Although these methods [37][36] have very steep timing roll-off metric, they are computationally intensive in number of operations required per point of timing metric.

Zhou [38][46] proposed a hybrid method for synchronization using preamble symbols. The method uses both kinds of correlation (auto and cross-correlation) operations with TS and multiple thresholds for detection of first path in multipath channel. But it uses both correlation operations together simultaneously, thus using a lot of computations per

timing metric point calculation. This leads to low throughput in output timing metric computation and hence can cause delay in synchronization. There is a need for a low complexity synchronizer which can quickly synchronize with incoming signal in a highly dispersive channel. Low complexity leads to lower resource count and low power which is required both for Wireless and Optical systems.

3.3 Proposed Hierarchical Low-Complexity Synchronizer for Wireless OFDM Systems

3.3.1 OFDM System Description

The transmitted baseband OFDM samples can be written in terms of IFFT equation as

$$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X[k] \cdot e^{j2\pi nk/N} \tag{3.1}$$

where N is the number of sub-carriers and $X[k]$ the complex information carrying symbol in frequency domain. The sampled signal at the receiver can be written as

$$r[n] = s[n-\eta] \cdot e^{j(2\pi\epsilon n/N+\phi)} + w[n] \tag{3.2}$$

where η is the integer timing offset, ϵ the CFO and ϕ the phase offset. $w[n]$ is the additive white Gaussian noise (AWGN) and $s[n]$ in multipath channel is given by

$$s[n] = \sum_{m=0}^{L_h-1} h[m] \cdot x[n-\tau_m] \tag{3.3}$$

where h is the sampled channel response (complex channel coefficients) at the receiver. L_h is the number of channel paths and τ_m is the path delay corresponding to the mth channel path. The channel is assumed static for the duration of the OFDM symbol.

To achieve synchronization in ISI channel at low complexity, a new synchronization method is proposed based on proposal of a new training symbol. The training symbol has low PAPR and can support both delay correlation and conjugate symmetry correlation operations. The training symbol is generated using CAZAC sequence, which have very low PAPR and possess impulse-like auto-correlation properties and constant cross-correlation property.

3.3.2 Proposed Hierarchical Method

The training sequence is based on modified Chu (CAZAC) [10] sequence, which have smaller alphabet size than Chu sequence. The modified Chu sequence [10] is given by

$$a_k^{(r)} = \begin{cases} exp\left(i\frac{2\pi}{N_s}\left\lfloor\frac{rk^2}{2}\right\rfloor\right), & \text{for } N_s \text{ even} \quad (3.4) \\ exp\left(i\frac{2r\pi k(k+1)}{N_s}\right), & \text{for } N_s \text{ odd} \quad (3.5) \end{cases}$$

where $0 \leq k < N_s$, $\gcd(r, N_s) = 1$ and $\lfloor a \rfloor$ denotes the integer part of a. Here $r = 1$ is used. The alphabet size is N_s for modified Chu sequence compared to $2N_s$ for Chu sequence.

The training symbol proposed [11] is

$$[\boldsymbol{C}\ \boldsymbol{C}\ \boldsymbol{C}\ -\boldsymbol{C}], \boldsymbol{C} = [\boldsymbol{A}\ \boldsymbol{B}], \boldsymbol{B} = \boldsymbol{A}^*[-\boldsymbol{n}]$$

The construction of repeating part $\boldsymbol{C}$ is done starting from $a_k^{(r)}$, which is considered to be in frequency domain. The size of $\boldsymbol{C}$ is $\frac{N}{8}$. The sequence generated is given to IFFT. At the input of IFFT, the zero frequency sub-carrier is switched off and as well as high-frequency sub-carriers, just like in LTE standard generation of primary synchronization signal (PSS) [47]. The output of IFFT is considered as $\boldsymbol{C}$ and is repeated to generate the proposed training sequence. The part $\boldsymbol{A}$ is constructed by taking IFFT of the generated modified Chu sequence [12] of size $N_s = \frac{N}{8}$. Then $\boldsymbol{B}$ is constructed from $\boldsymbol{A}$ by time-reversal and conjugation operation. The sign pattern $[1\ 1\ 1\ -1]$ is designed to ensure steep roll-off for initial estimation algorithm. The length of a single part is designed to be greater than maximum delay spread of the multipath channel. The initial timing metric proposed for the training symbol is delay based auto-correlation method that involves using repeating pattern for delay based correlation. The timing metric for coarse initial estimate is:

$$TM_{init}[n] = \left(\frac{L}{L-1} \cdot \frac{|P_{init}[n]|}{R_{init}[n]}\right)^2 \quad (3.6)$$

where P_{init} is the auto-correlation function, R_{init} is the energy calculation function, TM_{init} is the timing metric function and L is the number of repeating parts ($L = 4$) in the proposed training symbol. The term $\frac{L}{(L-1)}$ is used to normalized for maximum value of 1 at the correct starting point. The expressions for P_{init} and R_{init} are

$$P_{init}[n] = \sum_{k=0}^{L-2} u[k] \sum_{m=0}^{M-1} r^*[n + kM + m] \cdot r[n + (k+1)M + m] \quad (3.7a)$$

$$R_{init}[n] = \sum_{k=0}^{L-1} \sum_{m=0}^{M-1} |r[n + kM + m]|^2 \quad (3.7b)$$

where $u[k] = p[k] \cdot p[k+1]$, $p[k]$ contains the sign pattern of $[1\ 1\ 1\ -1]$, $k = 0, 1, ..., (L-1)$ and $M = N/L$. The time index corresponding to the maximum value gives the initial

estimate.

$$\widehat{\eta}_{init} = arg \max_{n} TM_{init}[n] \tag{3.8}$$

Figure 3.1a shows the plot of $TM_{init}[n]$ for a Signal-to-Noise Ratio (SNR) of 10 dB in a frequency selective channel. The maximum peak is not exactly at zero index which is the actual start of the OFDM symbol, but slightly shifted to the right due to multipath effect. The fine estimation algorithm consists of correcting this unknown shift and finding the correct starting point. The fine estimation algorithm does not assume a dominant first path and can work with non-dominant first path in multipath channel.

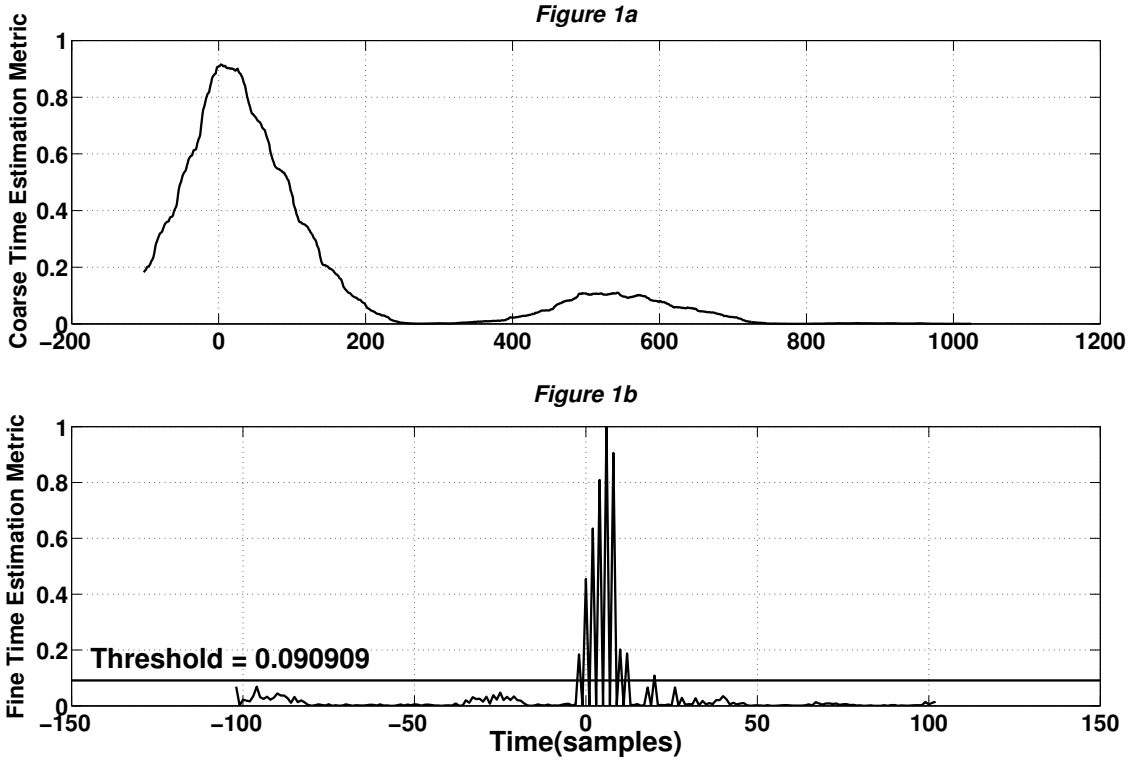

FIGURE 3.1: Plot of Coarse (a) and Fine (b) Timing Metric Functions

The fine estimation algorithm uses the conjugate symmetry present in the training symbol to estimate the correct starting point. The fine time estimation algorithm starts from the point $\widehat{\eta}_{center} = \widehat{\eta}_{init} + \frac{N}{2}$. Since the length of $\boldsymbol{A}$ or $\boldsymbol{B}$ part is greater than the maximum delay spread of the multipath signal, all the four parts of the training symbol are exposed to similar multipath channel environment. A search distance of $[-N_{cyp}, N_{cyp}]$ is covered from $\widehat{\eta}_{center}$ for finding all paths of multipath channel. The fine timing estimation metric is given by

$$TM_{fine}[n] = |P_{fine}[n]|^2 \tag{3.9}$$

with

$$P_{fine}[n] = \sum_{k=0}^{\frac{N}{4}-1} r[n-k-1] \cdot r[n+k] - \sum_{k=\frac{N}{4}}^{\frac{N}{2}-1} r[n-k-1] \cdot r[n+k] \tag{3.10}$$

where TM_{fine} is the fine timing estimation metric and P_{fine} is the conjugate symmetric correlation operation. The negative sign for $k \in [\frac{N}{4}, \frac{N}{2} - 1]$ is because of the sign pattern $[1\ 1\ 1\ \ -1]$, $n \in [-N_{cyp}, N_{cyp}]$. This range for n was chosen since the initial estimate does not produce peaks outside the maximum length of multipath channel. The timing metric produces peaks which are proportional to individual squared channel path gains. The expansion of $P_{fine}[n]$ in terms of channel coefficients is given by Equation 3.11.

$$\begin{aligned} P_{fine}[n] &= \sum_{k=0}^{M-1} r[n-k-1] \cdot r[n+k] \qquad (3.11)\\ &= \sum_{k=0}^{M-1} \left(\sum_{m=0}^{L_h-1} h_m x[n-k-1-\tau_m] + w[n-k-1]] \sum_{m'=0}^{L_h-1} h_{m'} x[n+k-\tau_{m'}] + w[n+k] \right) \\ &= \sum_{k=0}^{M-1} \sum_{\substack{m=0,\\ m=m'}}^{L_h-1} h_m^2 |x[n-k-1-\tau_m]|^2 + \sum_{k=0}^{M-1} \left(\sum_{m=0}^{L_h-1} \sum_{\substack{m'=0,\\ m' \neq m}}^{L_h-1} h_m h_{m'} x[n-k-1-\tau_m] \cdot x[n+k-\tau_{m'}] \right) \\ &\quad + W[n] \\ P_{fine}[n] &\approx \sum_{k=0}^{M-1} \sum_{\substack{m=0,\\ m=m'}}^{L_h-1} h_m^2 |x[n-k-1-\tau_m]|^2 \qquad (3.12) \end{aligned}$$

The term $W[n]$ refers to the correlation terms produced due to noise and signal components. The first term in Equation 3.11 corresponds to peaks produced in the timing metric. $Q[n]$ is calculated by normalizing all values of $TM_{fine}[n]$ by the maximum value of $TM_{fine}[n]$:

$$Q[n] = \frac{TM_{fine}[n]}{max(TM_{fine}[n])} \tag{3.13}$$

Figure 3.1b shows the plot of $Q[n]$ for SNR of 10 dB. The figure shows peaks corresponding to multipath gains. The threshold shown in 3.1b helps in selecting signal only components which are to be used for the windowed summation method to find the first arrival path. The time index of the maximum value of $Q[n]$

$$\widehat{\eta}_{fine} = arg \max_{n}(Q[n]) \tag{3.14}$$

is used as the starting point for windowing summation method which is similar to the one used in Choi's. The values of $Q[n]$ are thresholded by a value β.

$$Q[n] = \begin{cases} Q[n], & Q[n] > \beta, \\ 0, & otherwise, \end{cases} \tag{3.15}$$

β is the threshold which separates signal and noise components in $Q[n]$. This threshold is

determined using the probability distribution of the noise component in $Q[n]$. The steps are as follows:

- The sequence $Q[n]$ is passed through Lloyd-Max [13] quantization algorithm using three levels of quantization.
- The lowest quantization level and its cluster are considered as noise here. It is observed that this cluster follows a lognormal distribution. Mean (μ_n) and variance (σ_n^2) of noise cluster are calculated first. The corresponding mean μ and σ for lognormal distribution is

$$\mu = log\left(\frac{\mu_n^2}{\sqrt{\sigma_n^2 + \mu_n^2}}\right) \tag{3.16a}$$

$$\sigma = \sqrt{log\left(\frac{\sigma_n^2}{\mu_n^2} + 1\right)} \tag{3.16b}$$

- A constant false alarm rate (CFAR) of "α" is used for calculation of threshold. The equation for threshold is derived by integrating probability distribution function (pdf) of the noise distribution with limits [β, ∞].

$$\beta = e^{\left(\sqrt{2}\cdot\sigma\cdot erf^{-1}(1-2\cdot\alpha)+\mu\right)} \tag{3.17}$$

A constant false alarm rate is used across all SNR values. A windowed summation is performed after discarding the noise values using the threshold (β) calculated.

$$E_p(n) = \sum_{k=0}^{S_w-1} Q(\widehat{\eta}_{fine} - n + k) \tag{3.18}$$

where S_w is the length of summation window and J_m is the search window for signal component. Then the first arrival path is given by

$$\widehat{\eta}_{first} = arg\max_n E_p(n) : n = 0, 1, \cdots J_m \tag{3.19}$$

Finally,

$$\widehat{\eta}_{final} = \widehat{\eta}_{init} - \widehat{\eta}_{first} \tag{3.20}$$

This value indicates final estimate of the starting index of the OFDM symbol.

3.3.3 Carrier Frequency Offset (CFO) Estimation

Using estimated $\widehat{\eta}_{final}$ as the start of the training symbol, the negative sign in the training symbol is inverted to get [$\boldsymbol{C}$ $\boldsymbol{C}$ $\boldsymbol{C}$ $\boldsymbol{C}$]. There exists some interference due to minus sign in the channel response. The CFO estimate is calculated using the formula

$$\widehat{\epsilon}_{frac} = \frac{2}{\pi}\underline{/P[\eta_{final}]} \tag{3.21}$$

where $P[\eta_{final}]$ is the autocorrelation among the four parts of the training symbol. The calculation of $P[\eta_{final}]$ is done using (3.7a), difference being sign pattern of [1 1 1 1]. It gives fractional and integer CFO estimation. The integer CFO estimation range is equal to $\pm\frac{L}{2}$ sub-carrier spacing. The CFO estimation range of $\widehat{\epsilon}_{frac}$ for proposed TS is $\pm$ 2 sub-carrier spacing.

3.4 Simulation Results

3.4.1 Parameters

The performance of all synchronization algorithms has been investigated by using intensive Monte-Carlo simulations (10^5 runs). Algorithms using auto-correlation and cross-correlation techniques are compared here. The OFDM system parameters are shown in Table 3.1. The channel used here is a frequency selective channel (ISI channel) with an exponential Power Delay Profile (PDP) and ratio of first to last Rayleigh fading tap is set to 20 dB. A uniformly distributed random phase component is multiplied to every path during each simulation run. The channel has 16 taps with equal tap spacing of four samples. The windowing parameters used for Choi's synchronization method are $J = 41, S = 48$.

TABLE 3.1: Simulation Parameters

Parameters	Value
IFFT/FFT Size (N)	1024
Number of sub-carriers	1024
Length of Cyclic Prefix (N_{cyp})	102
OFDM Symbol Length (N_{sym})	1126
Window Size S_w (samples)	40
Distance J_m (samples)	36
Constant False alarm rate (α)	0.01
Number of simulation runs	10^5
Number of channel taps	16
Channel Tap Spacing (samples)	4
Ratio between first tap to last tap(in dB)	20
Carrier Frequency Offset (CFO)(ϵ)	0.75

3.4.2 Mean Square Error (MSE) of Timing Estimate

Figure 3.2 shows MSE of timing estimation in the ISI channel for various synchronization methods. The delay correlation based methods (Schmidl, Minn, Shi) have higher MSE compared to methods using only conjugate symmetry correlation (Park, Choi). Park's and Choi's methods estimate strongest channel path of the received multipath channel using conjugate symmetry correlation method. But only conjugate symmetry correlation does not give low MSE. Choi uses a windowed summation method for identifying first

path, which reduces the MSE compared to Park's method. Proposed method uses conjugate symmetry and windowed summation similar to Choi's to get low MSE in estimation which is better than Park and comparable to Choi at significantly lower computational complexity as shown in Figure 3.2.

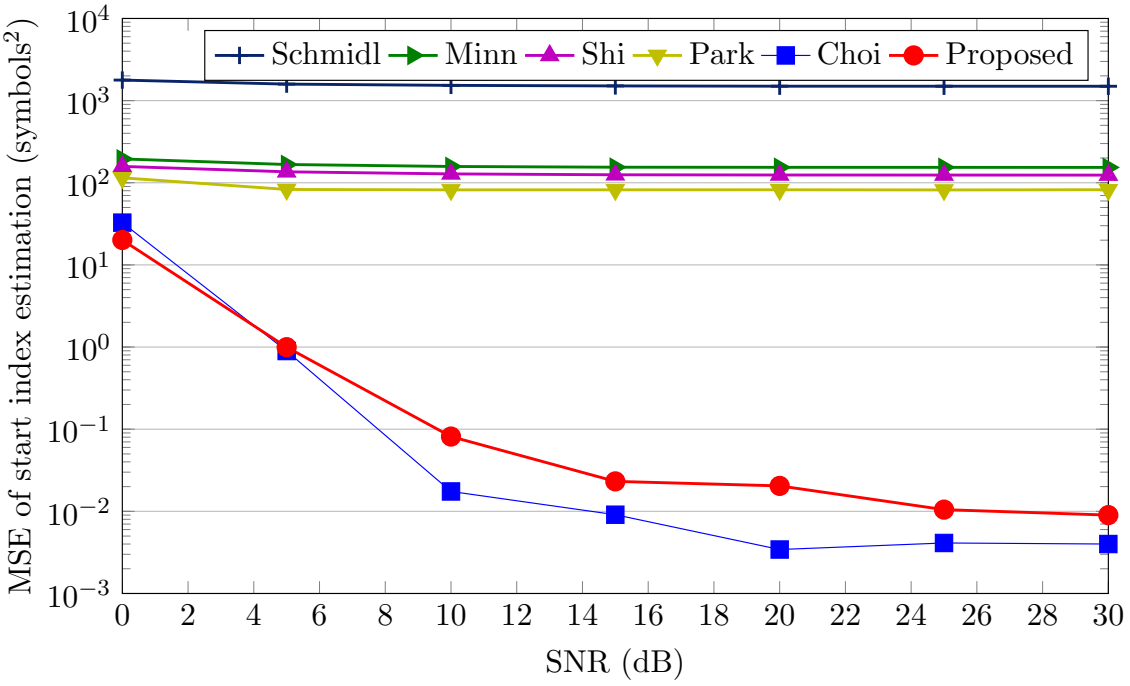

FIGURE 3.2: MSE of Timing Estimation versus SNR in ISI channel

3.4.3 Mean Square Error (MSE) of CFO Estimate

Figure 3.3 shows the MSE of CFO estimation for different SNR values. A CFO of 0.75 was used during simulation. No CFO estimation was proposed in Choi and Park's methods. The Cramér-Rao bound for variance in estimation of frequency offset [45] is given by

$$CRB(\hat{\epsilon}) = \frac{1}{2\pi^2} \frac{3(SNR)^{-1}}{N(1-1/N^2)} \tag{3.22}$$

where N is the size of FFT. Schmidl's CFO estimator comes closest to lower bound at all SNR values. Minn's CFO estimator uses algorithm of Morelli [45], which is computationally more complex compared to Schmidl's. Shi's CFO estimation algorithm hits a floor after SNR of 15 dB. Proposed CFO estimator is similar to Schmidl's and gives estimates very close to Schmidl's at medium to high SNR values because of more accurate estimation of the starting index compared to Schmidl's algorithm which does not suffer from interference from any negative sign in training symbol. The CFO estimator has a range of $\pm$ 2 sub-carrier spacing compared to Schmidl's which has a range of $\pm$ 1 sub-carrier spacing.

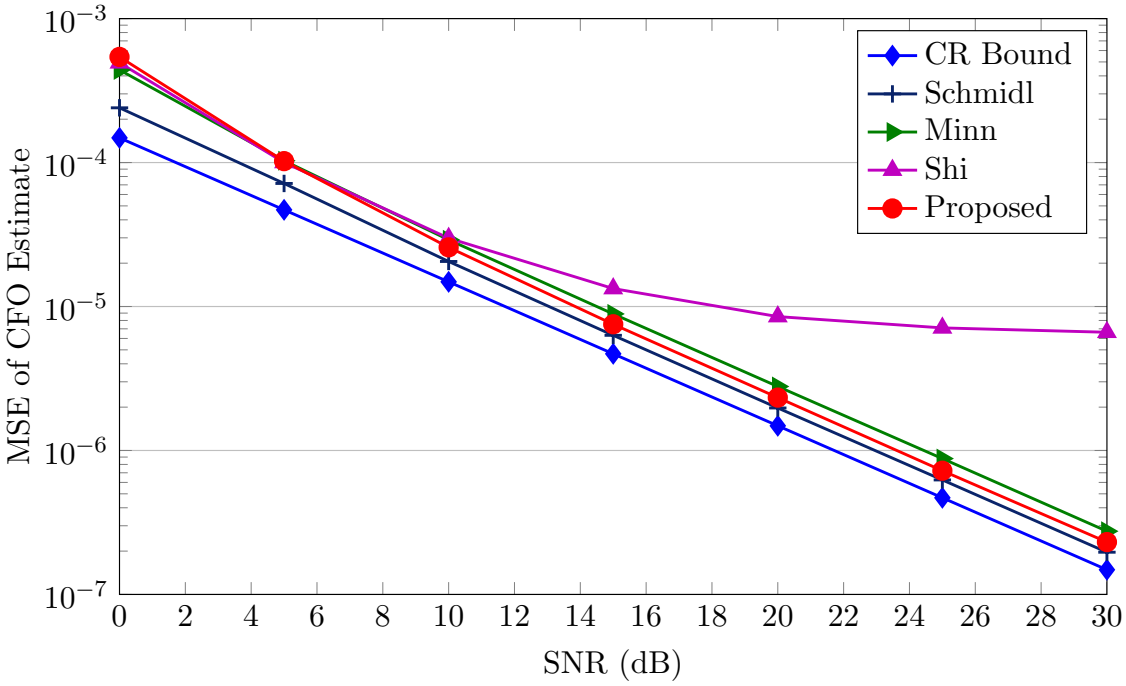

FIGURE 3.3: MSE of CFO Estimation versus SNR in ISI channel

3.4.4 Complexity of Calculations

The number of real operations required is given as a function of N in Table 3.2. Since the algorithms of Schmidl, Minn and Shi can be written in iterative form, the number of operations is fixed per timing metric calculation. But for algorithms of Park, Choi and Zhou where the numerator part cannot be written in the form of iterative formula, the number of operations becomes a function of N. Since the proposed algorithm has initial iterative part and fine non-iterative part, the fine estimation algorithm is a function of N. But, the non-iterative part works only for calculation of $(2N_{cyp}+1)$ samples, unlike cross-correlation algorithms which work on every OFDM symbol (N_{sym}). Real operations of cross-correlation algorithms and proposed fine step is $\approx 2N$, but since proposed fine step operates on lesser number of samples. For calculation of timing metric over one OFDM symbol (N_{sym}), the reduction in complexity is

$$\begin{aligned} \textit{Complexity Reduction} &= \frac{(N_{sym} - (2N_{cyp}+1)) \cdot 2N}{N_{sym} \cdot 2N} \\ &= \frac{N_{sym} - (2N_{cyp}+1)}{N_{sym}} \end{aligned} \tag{3.23}$$

This results in approximately 80% reduction in computations using numerical values $(N_{sym} = 1126, N_{cyp} = 102)$ of simulation compared to Choi, Park and Zhou, while MSE performance is very close to Choi's MSE. In case of Minn, to perform the fine timing estimation, Maximum Likelihood (ML) channel estimation is done first which requires too

TABLE 3.2: Number of Real Operations for calculation of a single timing metric point

Algorithm	Real Multiplication	Real Addition	Division
Schmidl-Cox	15	13	1
Minn($L = 4$)	31	29	1
Shi	59	61	1
Park	$(2N + 11)$	$(2N + 7)$	1
Choi	$(2N + 7)$	$(2N + 3)$	1
Zhou	$(2N + 22)$	$(2N + 16)$	1
Proposed coarse step ($L = 4$)	31	29	1
Proposed fine step	$(2N + 3)$	$(2N - 1)$	1

much complexity. In case of Schmidl, the MSE is very high although it is computationally efficient. So, in terms of computational complexity, the proposed algorithm is significantly better than Choi, Park and Zhou and in terms of MSE, significantly better than Schmidl, Minn and Shi's methods. Proposed method is specially useful for MIMO systems where each antenna calculates timing synchronization and hence it needs to be computationally and resource efficient. Thus, the proposed algorithm provides a very good trade-off between computational complexity and MSE of timing and CFO estimation in a frequency selective channel.

3.5 Hierarchical Synchronizer Proposed for CO-OFDM System

The proposed hierarchical low-complexity synchronizer for Wireless channel OFDM systems has three parts for achieving low value of MSE, namely

- Auto-correlation Operation (TM_{init}, Eq. 3.6)
- Conjugate Symmetric Correlation Operation (TM_{fine}, Eq. 3.9)
- Windowed Summation (E_p, Eq. 3.18)

In case of SMF optical channel, the dispersion value is not high and more stable compared to multi-path effect of the wireless channel which can have large delay. Hence, the windowed summation step in the proposed synchronizer can be eliminated and only a 2- step procedure is necessary. This modified algorithm is used for synchronization in SMF optical channel. For comparison purposes, only auto-correlation algorithms (Schmidl-Cox, Minn-Bhargava, Shi-Serpedin) are compared. Cross-correlation algorithms (Choi, Park) are not compared since they require large amount of resources and do not provide output every cycle. Also, cross-correlation algorithms do not provide CFO estimation. The hierarchical

synchronization steps used to calculate starting point of OFDM symbol $\eta_{final} = \eta_{init} - \eta_{fine}$ are:

- Auto-Correlation Operation - Eq. 3.6 is used without change to calculate η_{init}.
- Conjugate Symmetric Correlation Operation - Eq. 3.9 is modified to reduce the complexity and normalization is done by using the energy of the symbol.

$$\eta_{fine} = arg \max_{n} \left(TM^{lc}_{fine}[n] \right) \tag{3.24}$$

$$TM^{lc}_{fine}[n] = \frac{|P^{lc}_{fine}[n]|^2}{R^2_{fine}} \tag{3.25}$$

$$P^{lc}_{fine}[n] = \sum_{k=0}^{\frac{N}{4}-1} r[n-k-1] \cdot r[n+k] \tag{3.26}$$

$$R_{fine}[n] = \sum_{k=0}^{\frac{N}{4}-1} |r[n+k]|^2 \tag{3.27}$$

The CFO estimation ($\widehat{\epsilon}_{frac}$) is done using Eq. 3.21.

3.6 Simulation Results

3.6.1 Parameters

The performance of all synchronization algorithms has been investigated by using intensive Monte-Carlo simulations (10^4 runs). The optical fiber channel simulated was standard single mode fiber (SSMF). The SSMF was simulated in Matlab using Optilux [48] simulator. Optilux Simulator helps in simulation of each component of the optical system with realistic parameter values and non-idealities modeled. Simulation parameters are given in Table 3.3. Simulation is done first with just fractional CFO and then with integer CFO included to see the sensitivity of the algorithms to large CFO value.

3.6.2 MSE of Timing Estimate

Figure 3.4 shows the plot of MSE of timing estimation in SSMF channel with CFO = 0.75 sub-carrier spacing. It can be observed for the proposed algorithm that there is a degradation in performance at low OSNR due to reduced computational complexity in calculation of P^{lc}_{fine}. Proposed algorithm's MSE curve shows improvement with increasing OSNR. Figure 3.5 shows the plot of MSE of timing estimation vs OSNR with CFO = 4.75. Performance of all the algorithms is similar to the case of only fractional CFO, showing proposed algorithm incurs no performance penalty in case of large values of CFO.

TABLE 3.3: Simulation Parameters for CO-OFDM System Simulation

OFDM Parameters	Value
IFFT/FFT Size (N)	256
Number of sub-carriers	256
Length of Cyclic Prefix (N_{cyp})	8
OFDM Symbol Length (N_{sym})	264
Number of simulation runs	10^4
DAC Sampling Frequency ($F_{s,DAC}$)	5 GHz
SSMF Channel Parameters	**Value**
Length of Fiber (L_F)	1000 km
Fiber Attenuation (α_{dB})	0.2 dB/km
Fiber Effective Area (A_{eff})	80 μm^2
Lambda (λ)	1550 nm
Chromatic Dispersion (CD)	17 $ps/(nm-km)$
Polarization Mode Dispersion (PMD)	0.05 $ps/\sqrt{km}$
Fiber Slope	0.089 $ps/(nm^2-km)$
Differential Group Delay (DGD)	12.5
Carrier Frequency Offset (CFO)(ϵ)	0.75 and 4.75

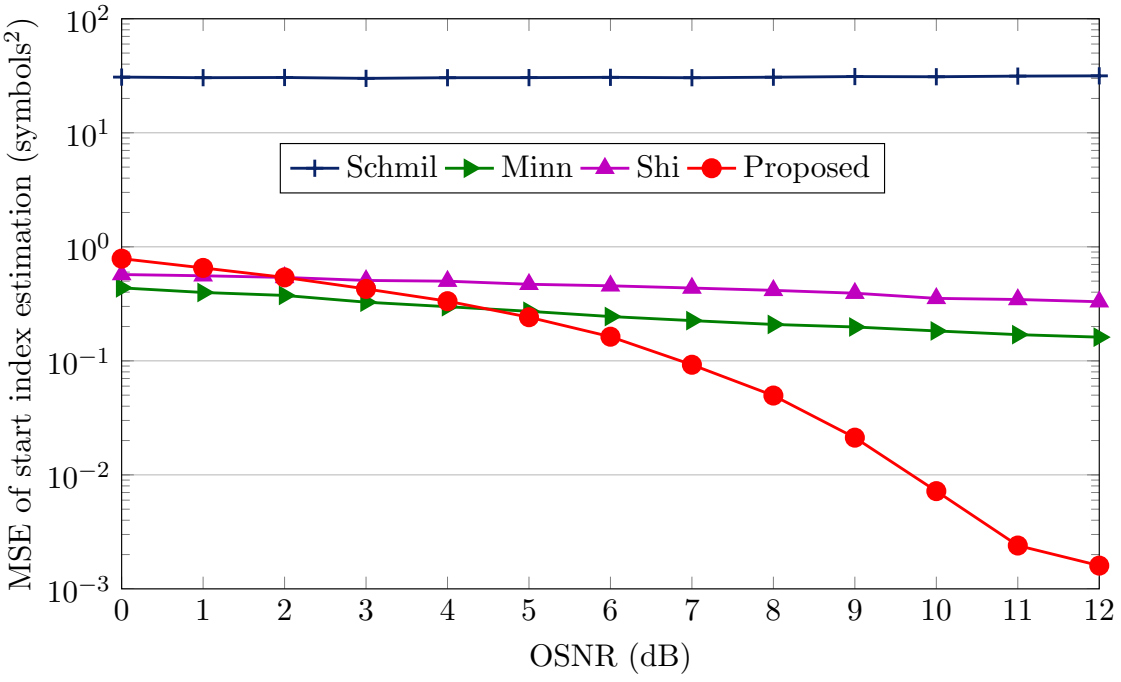

FIGURE 3.4: MSE of Timing Estimation vs. OSNR in SSMF channel with CFO = 0.75

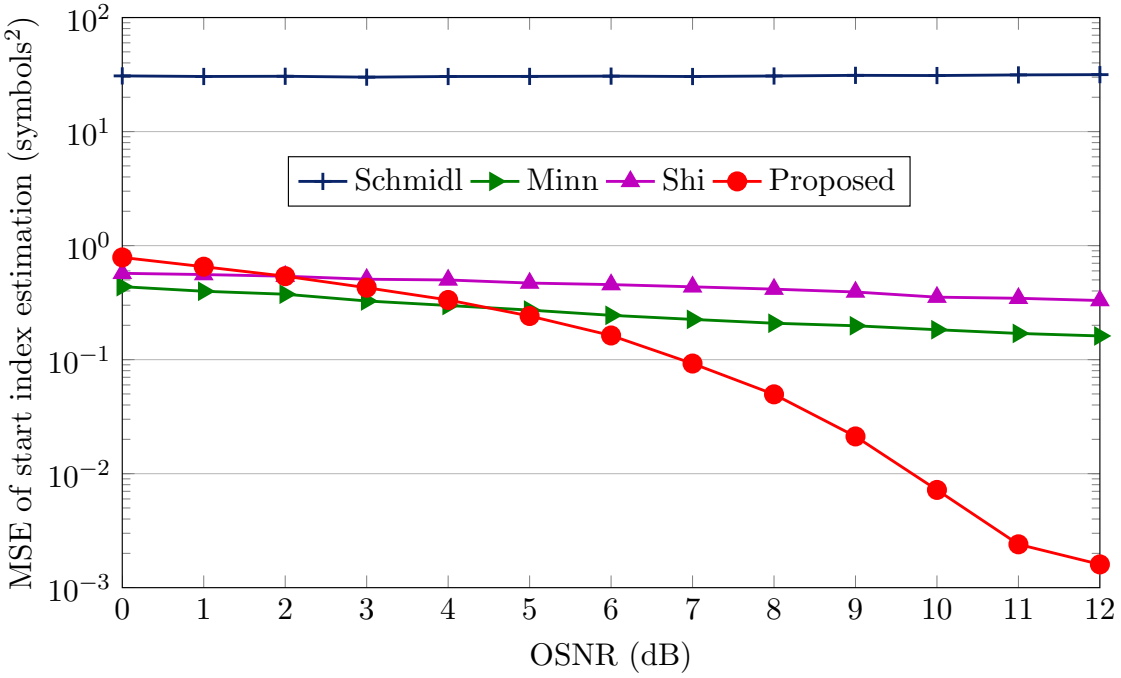

FIGURE 3.5: MSE of Timing Estimation vs. OSNR in SSMF channel with CFO = 4.75

3.6.3 MSE of CFO Estimate

Figure 3.6 shows the MSE of CFO estimation versus OSNR for CFO = 0.75. The MSE of CFO estimation is close to Schmidl-Cox algorithm.

3.7 Need for Parallel Timing Synchronization Architecture

In this section, architectures for timing synchronization algorithms are explored in the context of CO-OFDM systems. In a typical CO-OFDM system, the input serial rate is of the order of Gb/s. The Gb/s serial input rate is provided by ADC, which operates at a sampling rate (F_s) of GHz. When input serial data rate is given to deserializer block of FPGA, it converts the high rate to lower rate which is a multiple of serial rate. The deserializer block now provides multiple samples per cycle at a lower frequency to FPGA, whose maximum frequency (F_{clk}) of operation can reach around 400 MHz. The ratio of $\frac{F_s}{F_{clk}}$ has ratio of 4, 8 or 16 [14], which makes it necessary for the OFDM receiver blocks to either process multiple parallel samples per cycle or to have multiple blocks which process single sample per cycle. The latter idea is resource intensive and does not scale well. The approach of CO-OFDM receiver blocks which can process multiple parallel samples per cycle and match the input is a scalable option which also offers resource savings by sharing among multiple parallel blocks.

In CO-OFDM system, the timing synchronization block is the first block in the receiver chain, which detects start of OFDM symbol. It also provides fractional CFO estimation for

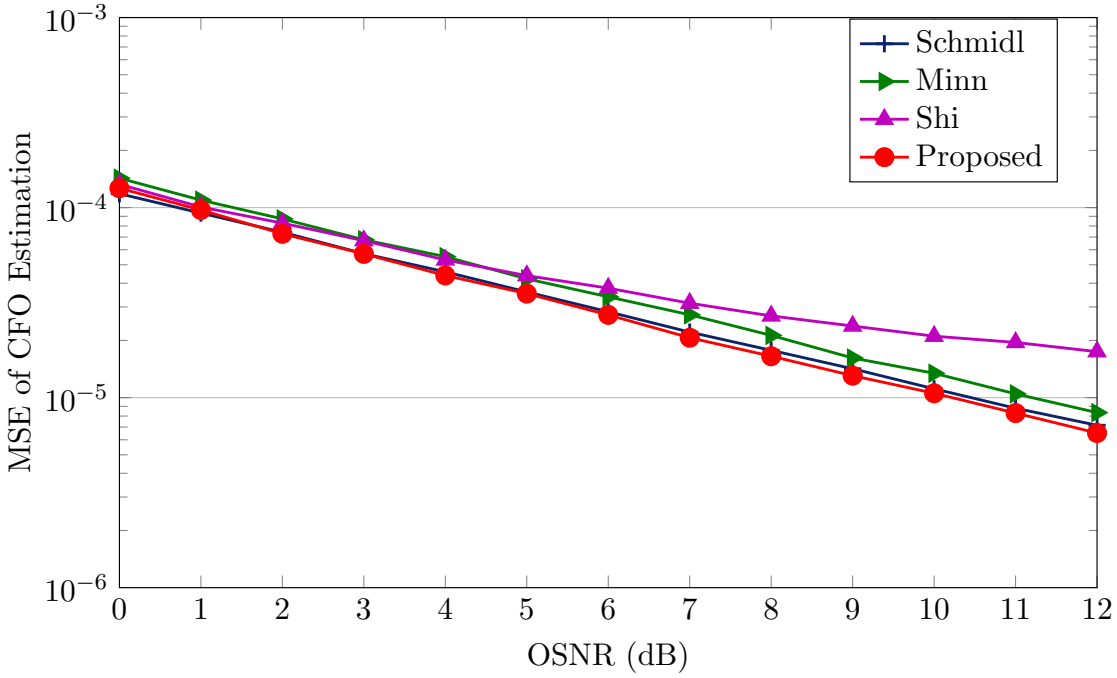

FIGURE 3.6: MSE of CFO Estimation vs. OSNR in SSMF channel for CFO = 0.75

the CFO compensation block. The block operates continuously to detect training symbols repeatedly sent at the beginning of each frame and tracks CFO variations. The total processing rate of the timing synchronization block has to match the input serial rate to avoid large memory for storing incoming data. In the present literature, there has been only two proposals for real-time parallel processing timing synchronization architecture. The details of the previous proposals are given below. Both proposals are based on using sample level parallelism to provide multiple outputs to match the input rate.

- Kaneda et al. [14] proposed an architecture by directly parallelizing the non-iterative auto-correlation equation for Schmidl-Cox algorithm, using training symbol was of the form $[A\ A]$:

$$P_{sc}^{kan}[n] = \sum_{k=0}^{M_{sc}/R\,R(k+1)-1} \sum_{m=0} r^*[n+m] \cdot r[n+m+M_{sc}] \tag{3.28}$$

  where M_{sc} is the length of repeating part of the training symbol (A), R is the number of parallel inputs and outputs, r is the input data stream, P_{sc}^{kan} is the auto-correlation function. Figure 3.7 shows the parallel architecture resulting from Eq. 3.28. For the case of $R = 16$ and $M_{sc} = 32$, it requires 64 real multipliers and 544 real adders. Hence, it is not an efficient parallel realization of the algorithm. The ratio of $\frac{F_s}{F_{clk}}$ was 16, but only one pipeline was realized and the training symbol was duplicated 16 times. This resulted in reduced spectrum efficiency and less accurate estimation

of starting point. The fractional CFO estimation was significantly reduced due to duplication of training symbol.

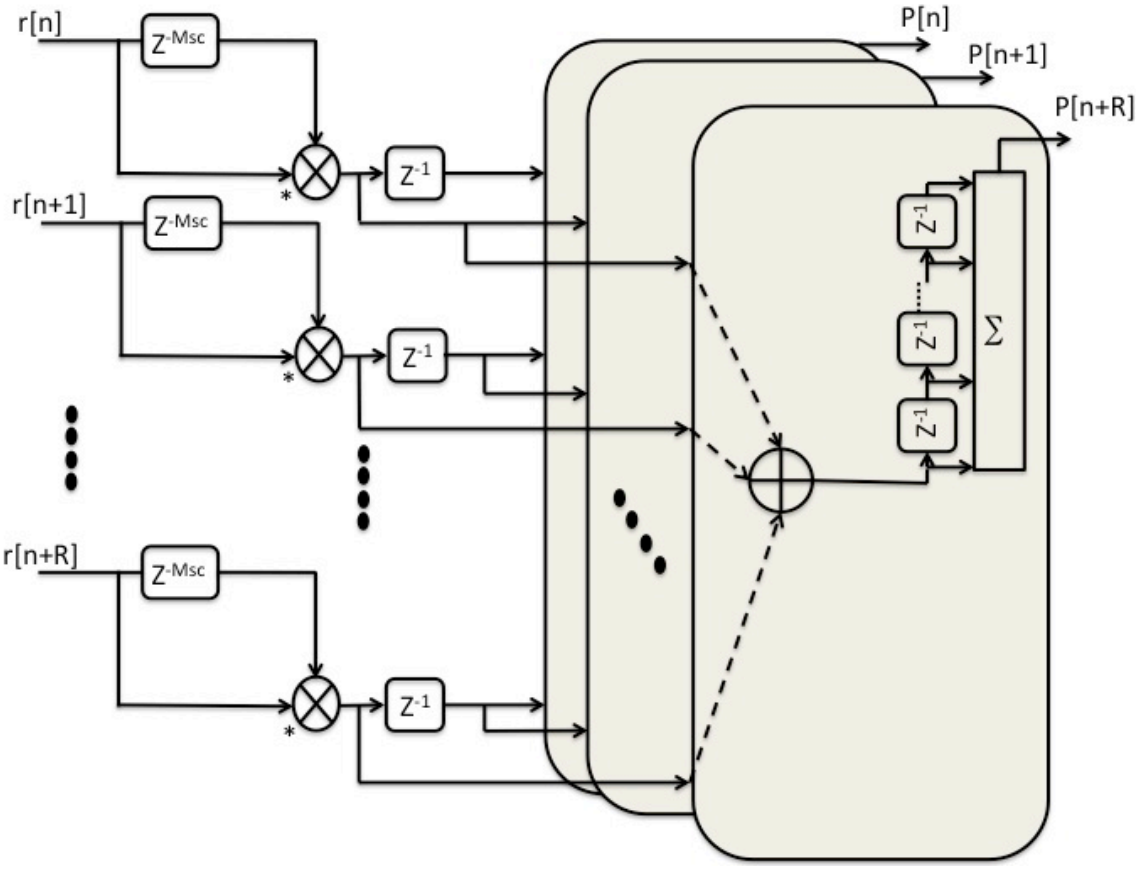

FIGURE 3.7: Parallel Architecture proposed by Kaneda et. al for Schmidl-Cox Algorithm

- Chen et al. [32] proposed an 8-parallel architecture which uses cross-correlation operation, and training symbol of the form $[A\ A\ A\ -A]$. The length of A was $M_{mb} = 32$. Although, complex multipliers were avoided, the number of adders required was large and it does not produce output every cycle unlike auto-correlation. Cross-correlation with known training symbol in presence of large CFO value results in shifted peaks, which reduces the accuracy of start point estimation. For $R = 8$-parallel architecture proposed by Chen et al. 2032 adders were required, and does not provide fractional CFO estimation.

Architectural complexity of the two architectures is shown in Table 2.9 for different parallel inputs/outputs. Previously proposed sample-level parallel architectures consume too much of resources and do not scale efficiently. Due to complexity of architecture, they do not provide fractional CFO estimation. The two proposals indicate that further efficiency improvement of symbol synchronization in parallel processing is required [14]. In the next sections, block-parallel architectures are proposed for acceleration of auto-correlation

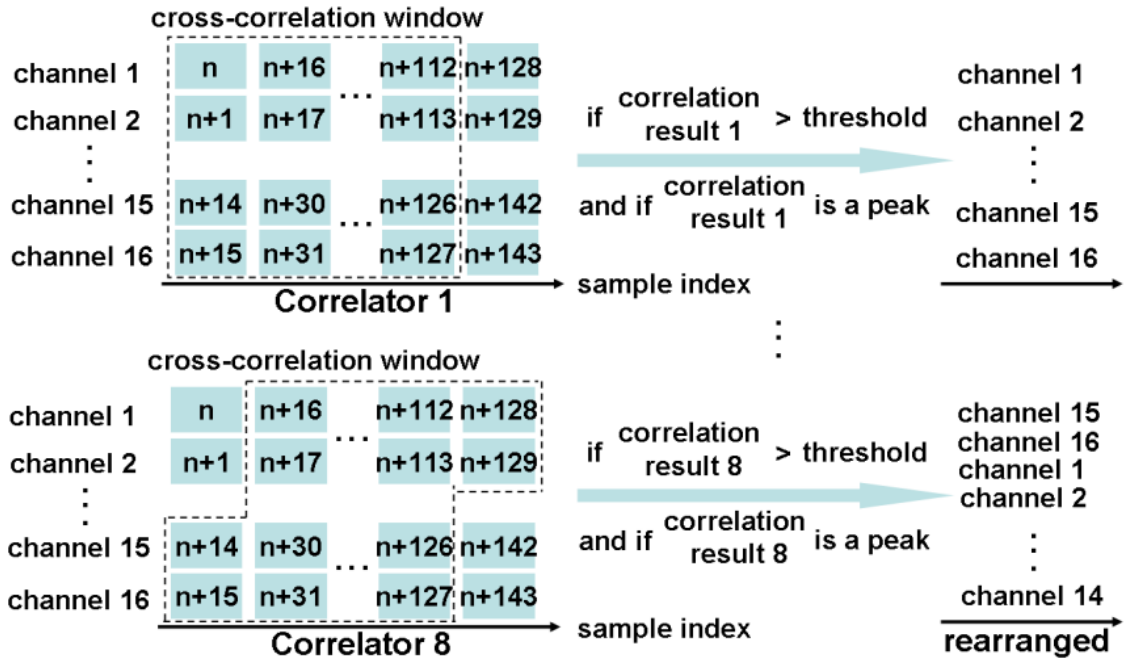

FIGURE 3.8: Parallel Architecture proposed by Chen et. al for cross-correlation operation

operation. Further, the proposed architecture is shown to accelerate conjugate symmetric correlation operation. Hence, it could accelerate both auto-correlation and conjugate symmetric correlation operation, thus helping in implementing the proposed hierarchical synchronization algorithm.

3.8 Proposed Block Parallel Architecture for Auto-Correlation

The goal of efficient parallelization of auto-correlation algorithm is to have a scalable architecture for efficiently handling multiple parallel inputs and match the input data rate. Block-level parallelism is proposed for achieving a scalable architecture compared to sample-level parallel approach. Proposed block-level architecture uses both non-iterative and iterative forms of equations to achieve parallel computation. Non-iterative equation is used to calculate auto-correlation for the first sample point of the blocks, while iterative equation is used for auto-correlation sample calculation for the remaining points in the block. The choice of block size for parallel computation determines the possible sharing of resources. The proposed block parallel method is explained as follows using Schmidl-Cox algorithm (SCA) and Minn-Bhargava algorithm (MBA) auto-correlation equations. For SCA, with a training symbol $[A\ A]$, the non-iterative and iterative auto-correlation

equations are as follows:

$$P_{sc}[n] = \sum_{m=0}^{M_{sc}-1} r^*[n+m] \cdot r[n+m+M_{sc}] \tag{3.29}$$

$$P_{sc}[n] = P_{sc}[n-1] + r^*[n+M_{sc}] \cdot r[n+2M_{sc}] - r^*[n] \cdot r[n+M_{sc}] \tag{3.30}$$

where Eq. 3.29 is non-iterative equation and Eq. 3.30 the iterative equation. P_{sc} is auto-correlation function, $M_{sc} = \frac{N}{2}$ is the size of repeating symbol (A). It can be observed that non-iterative correlation is time-consuming and does not produce outputs every cycle, while iterative equation can produce outputs every clock cycle, but depends on availability of past sample auto-correlation value. Since the non-iterative equation only depends on inputs, it can be used to calculate auto-correlation value which can be fed into iterative equation computation. Block-level parallelism [49] uses this idea and applies to multiple parallel blocks working in this fashion. Block size is important to increase sharing of resources. Observation of auto-correlation operation indicates its dependency of samples delayed by M_{sc} samples. This dependency can be used to decide the size of block to ensure maximum resource sharing among multiple parallel blocks. If the non-iterative and iterative equations are written for $R = 4$-parallel computation separated by M_{sc} samples apart, it is given by

$$P_{sc}[n] = \sum_{m=0}^{M_{sc}-1} r^*[n+m] \cdot r[n+m+M_{sc}] \tag{3.31}$$

$$P_{sc}[n+M_{sc}] = \sum_{m=0}^{M_{sc}-1} r^*[n+m+M_{sc}] \cdot r[n+m+2M_{sc}] \tag{3.32}$$

$$P_{sc}[n+2M_{sc}] = \sum_{m=0}^{M_{sc}-1} r^*[n+m+2M_{sc}] \cdot r[n+m+3M_{sc}] \tag{3.33}$$

$$P_{sc}[n+3M_{sc}] = \sum_{m=0}^{M_{sc}-1} r^*[n+m+3M_{sc}] \cdot r[n+m+4M_{sc}] \tag{3.34}$$

$$\begin{aligned} P_{sc}[n+1] = P_{sc}[n] &+ r^*[n+M_{sc}] \cdot r[n+2M_{sc}] \\ &- r^*[n] \cdot r[n+M_{sc}] \end{aligned} \tag{3.35}$$

$$\begin{aligned} P_{sc}[n+M_{sc}+1] = P_{sc}[n+M_{sc}] &+ r^*[n+2M_{sc}] \cdot r[n+3M_{sc}] \\ &- r^*[n+M_{sc}] \cdot r[n+2M_{sc}] \end{aligned} \tag{3.36}$$

$$\begin{aligned} P_{sc}[n+2M_{sc}+1] = P_{sc}[n+2M_{sc}] &+ r^*[n+3M_{sc}] \cdot r[n+4M_{sc}] \\ &- r^*[n+2M] \cdot r[n+3M_{sc}] \end{aligned} \tag{3.37}$$

$$\begin{aligned} P_{sc}[n+3M_{sc}+1] = P_{sc}[n+3M_{sc}] &+ r^*[n+4M_{sc}] \cdot r[n+5M_{sc}] \\ &- r^*[n+3M] \cdot r[n+4M_{sc}] \end{aligned} \tag{3.38}$$

It can be observed that, in the block-level parallel non-iterative equation computation the data retrieved from memory can be shared across computations and memory access is regular. In case of block-level parallel iterative equation, multiplier outputs can be shared which leads to significant decrease in multipliers used. Due to selection of block size of M_{sc}, this sharing has been made possible. Consider now the auto-correlation equations of Minn-Bhargava algorithm for training symbol pattern $[A\ A\ A\ -A]$,

$$P_{mb}[n] = \sum_{k=0}^{L-2} p[k] \cdot p[k+1] \sum_{m=0}^{M_{mb}-1} r^*[n+m+kM_{mb}] \cdot r[n+m+(k+1)M_{mb}] \tag{3.39}$$

$$P_{mb}[n] = P_{mb}[n-1] + r^*[n] \cdot r[n+M_{mb}] - r^*[n+3M_{mb}] \cdot r[n+4M_{mb}] + 2r^*[n+2M_{mb}] \cdot r[n+3M_{mb}] \tag{3.40}$$

where Eq. 3.39 is the non-iterative equation and Eq. 3.40 is the iterative equation, P_{mb} is the auto-correlation function, M_{mb} is the size of the repeating part (A), $M_{mb} = \frac{N}{4}$. If the non-iterative and iterative equations are written for $R = 4$-parallel computation using block-size of M_{mb}, it is given by

$$P_{mb}[n] = \sum_{k=0}^{L-2} p[k] \cdot p[k+1] \sum_{m=0}^{M_{mb}-1} r^*[n+m+kM_{mb}] \tag{3.41}$$

$$P_{mb}[n+M_{mb}] = \sum_{k=0}^{L-2} p[k] \cdot p[k+1] \sum_{m=0}^{M_{mb}-1} r^*[n+m+(k+1)M_{mb}] \tag{3.42}$$

$$P_{mb}[n+2M_{mb}] = \sum_{k=0}^{L-2} p[k] \cdot p[k+1] \sum_{m=0}^{M_{mb}-1} r^*[n+m+(k+2)M_{mb}] \tag{3.43}$$

$$P_{mb}[n+3M_{mb}] = \sum_{k=0}^{L-2} p[k] \cdot p[k+1] \sum_{m=0}^{M_{mb}-1} r^*[n+m+(k+3)M_{mb}] \tag{3.44}$$

$$P_{mb}[n+1] = P_{mb}[n] - r^*[n] \cdot r[n+M_{mb}] - r^*[n+3M_{mb}] \cdot r[n+4M_{mb}] + 2r^*[n+2M_{mb}] \cdot r[n+3M_{mb}] \tag{3.45}$$

$$P_{mb}[n+1] = P_{mb}[n] - r^*[n] \cdot r[n+M_{mb}] - r^*[n+3M_{mb}] \cdot r[n+4M_{mb}] + 2r^*[n+2M_{mb}] \cdot r[n+3M_{mb}] \tag{3.46}$$

$$P_{mb}[n+1] = P_{mb}[n] - r^*[n] \cdot r[n+M_{mb}] - r^*[n+3M_{mb}] \cdot r[n+4M_{mb}] + 2r^*[n+2M_{mb}] \cdot r[n+3M_{mb}] \tag{3.47}$$

$$P_{mb}[n+1] = P_{mb}[n] - r^*[n] \cdot r[n+M_{mb}] - r^*[n+3M_{mb}] \cdot r[n+4M_{mb}] + 2r^*[n+2M_{mb}] \cdot r[n+3M_{mb}] \tag{3.48}$$

It can be observed that due to selection of block size M_{mb}, which is the distance of auto-correlation, significant sharing of resources can be obtained. Based on sharing of

resources for non-iterative and iterative equations, two architectures are proposed which can support block-parallel computation.

3.9 Partial-Streaming Block-Parallel (PSBP) Architecture

Proposed architecture uses the shares resources completely between non-iterative and iterative equations. The architecture has two modes of operation, one for non-iterative mode which is done for initial auto-correlation point calculation, and the other one for iterative mode which is subsequent points in the block. Consider the computation of R M_{sc} auto-correlation outputs using R parallel PSBP architecture. Dividing the total computation work into R parallel blocks, each block computes $\frac{R\,M_{sc}}{R}$ outputs. The order of computation is as follows:

- Initially R initial auto-correlation points are calculated in non-iterative mode. It requires M_{sc} cycles in case of SCA and M_{mb} in case of MBA for completing the operation.
- The remaining $(M_{sc}-1)$ in case of SCA or $(M_{mb}-1)$ in case of MBA is computed in iterative mode. It can output $(M_{sc}-1)$ outputs in the same number of cycles and gives a throughput of R samples/cycle.

After the computation of R M_{sc} outputs, the same process is repeated for the calculation of next set of R M_{sc} outputs. The architecture consumes $(2M_{sc}-1)$ cycles for computing M_{sc} auto-correlation outputs per block. The architecture is called partial streaming because, it does not produce output every cycle and has a delay when computing initial auto-correlation output using non-iterative equation. The outputs are not fully streaming, but it uses minimum set of resources for computation. The following subsections depict the architecture for SCA and MBA.

3.9.1 Proposed PSBP architecture for Schmidl-Cox algorithm (SCA)

Figure 3.9 shows the $R=4$-parallel PSBP architecture for auto-correlation calculation in case of SCA. Energy calculation of SCA written in both non-iterative and iterative form are given by:

$$R_{sc}[n] = \sum_{m=0}^{M_{sc}-1} |r[n+m+M_{sc}]|^2 \tag{3.49}$$

$$R_{sc}[n] = R_{sc}[n-1] + |r[n+2M_{sc}]|^2 - |r[n]|^2 \tag{3.50}$$

Figure 3.10 shows the $R=4$-parallel PSBP architecture for energy calculation in case of SCA. Architectural complexity of the proposed PSBP architecture as a function of R parallel inputs/outputs is shown in Table 3.4. Resource requirement scales only as a function of R-parallel input/output and is independent of M_{sc}. It is compared with Kaneda's

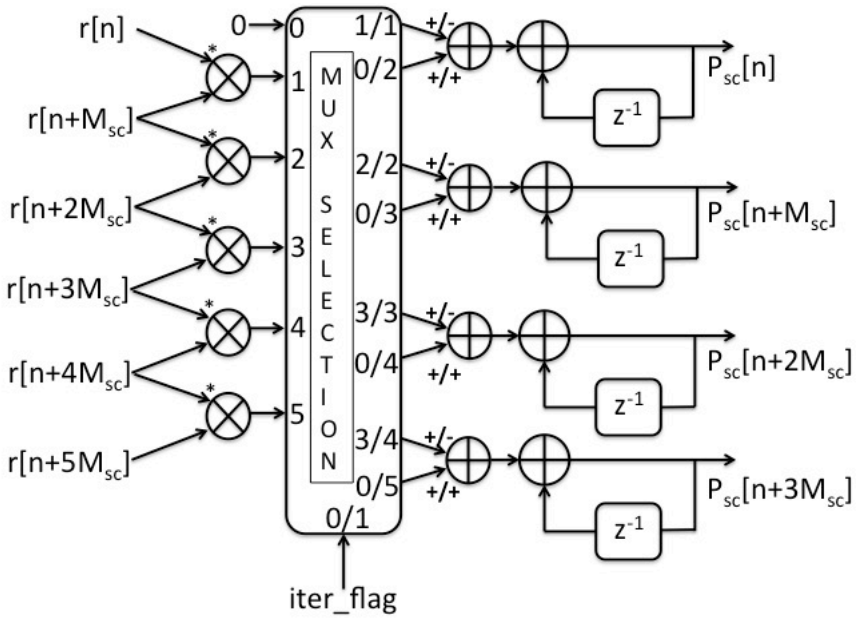

FIGURE 3.9: Proposed $R = 4$-Parallel PSBP Architecture for P_{sc} calculation in case of SCA. iter_flag = 0 indicates non-iterative computation mode, while iter_flag = 1 indicates iterative computation mode.

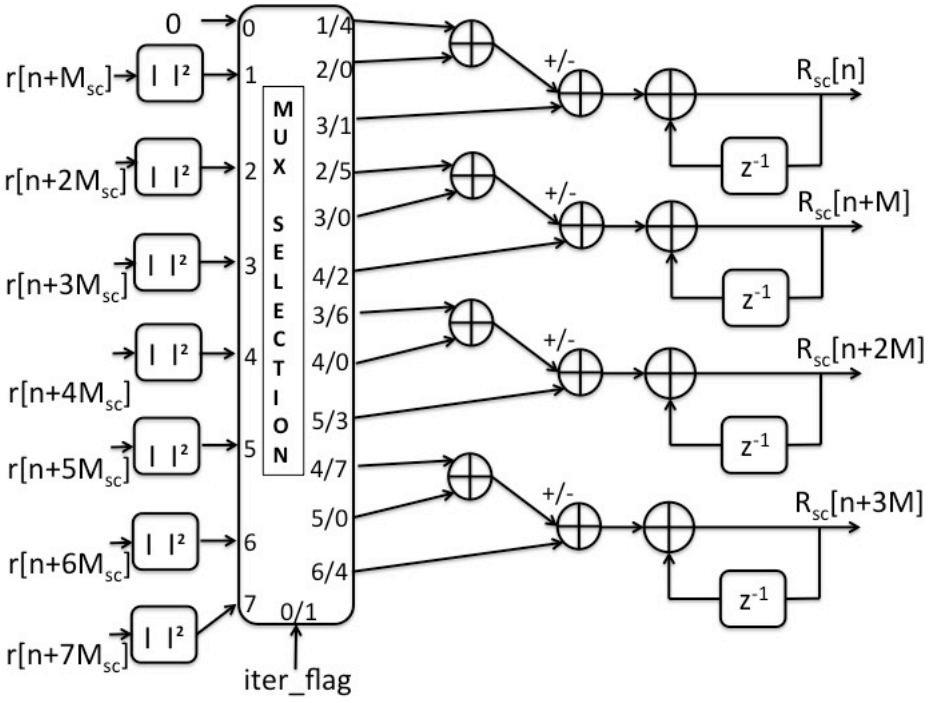

FIGURE 3.10: Proposed $R = 4$-Parallel PSBP Architecture for R_{sc} calculation in case of SCA. iter_flag = 0 indicates non-iterative computation mode, while iter_flag = 1 indicates iterative computation mode.

TABLE 3.4: Architectural Complexity calculation as a function of R-parallel input/output for SCA

Proposed architecture for Schmidl-Cox

Algorithm	Real Multipliers	Real Adders
P_{sc}	$4(R+1)$	$2(3R+1)$
R_{sc}	$2(R+1)$	$3R+1$

Kaneda's architecture for Schmidl-Cox

Algorithm	Real Multipliers	Real Adders
P_{sc}	$4R$	$2R(M_{sc}/2+1)$

proposal for SCA. It can be seen that proposed PSBP architecture requires four more multipliers compared to Kaneda's proposal, but this difference is fixed and independent of R. Number of adders required by Kaneda's architecture is large due to its dependence on size of training symbol (M_{sc}). For $M_{sc} = 32$, adder savings of proposed architecture for $R = 16$-parallel architecture is around 82%. Similar savings can be obtained for the architecture of R_{sc} compared to Kaneda's method of sample-level parallelization.

3.9.2 Proposed PSBP architecture of Minn-Bhargava algorithm (MBA)

Figure 3.11 shows the $R = 4$-parallel PSBP architecture for auto-correlation calculation in case of MBA. Energy calculation of MBA algorithm is given by

$$R_{mb}[n] = \sum_{k=1}^{L-2} \sum_{m=0}^{M_{mb}-1} |r[n+m+kM_{mb}]|^2 \tag{3.51}$$

$$R_{mb}[n] = R_{mb}[n-1] + |r[n+4M_{mb}]|^2 - |r[n+M_{mb}]|^2 \tag{3.52}$$

Figure 3.12 shows the $R = 4$-parallel PSBP architecture for energy calculation in case of MBA. Table 3.5 shows the architectural complexity as a function of R-parallel input/output for Minn-Bhargava algorithm. It is again compared with Kaneda's proposal for Schmidl-Cox. Proposed architecture requires twelve more real multipliers compared to Kaneda's proposal and it is fixed independent of R. This is because Minn-Bhargava auto-correlation (P_{mb}) has inherently higher number of computations compared to Schmidl-Cox auto-correlation (P_{sc}). But, there are savings in area required compared to Kaneda's proposal. For $R = 16$-parallel output, adder savings for proposed architecture is around 81%. Similar savings can be obtained for the architecture of R_{mb} compared to Kaneda's method of sample-level parallelization.

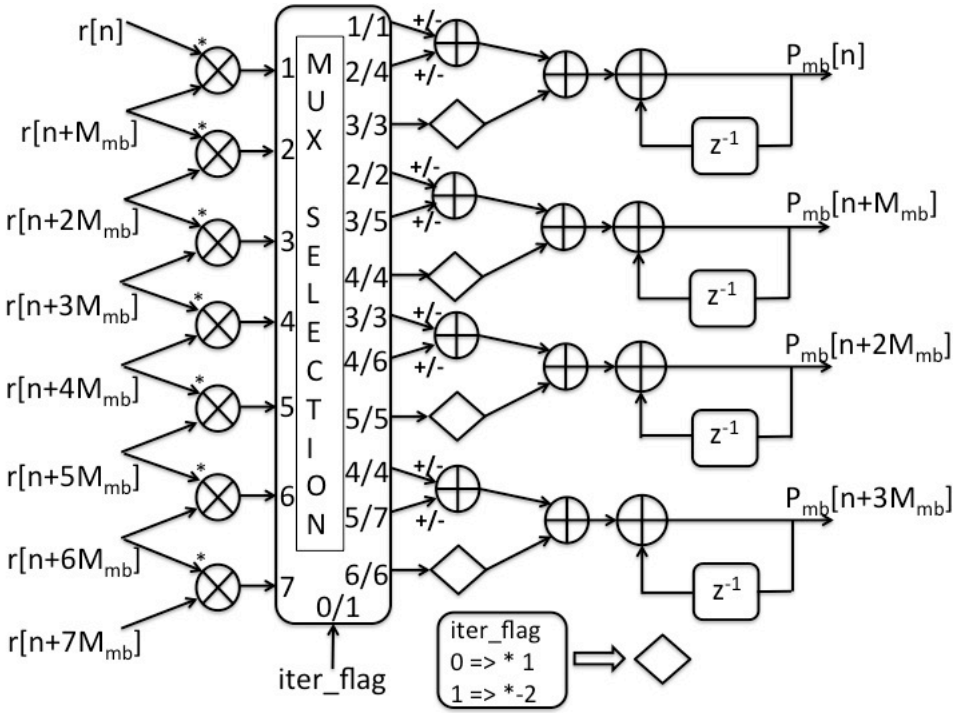

FIGURE 3.11: Proposed PSPB Architecture for calculation of P_{mb} in case of MBA. iter_flag = 0 indicates non-iterative computation mode, while iter_flag = 1 indicates iterative computation mode.

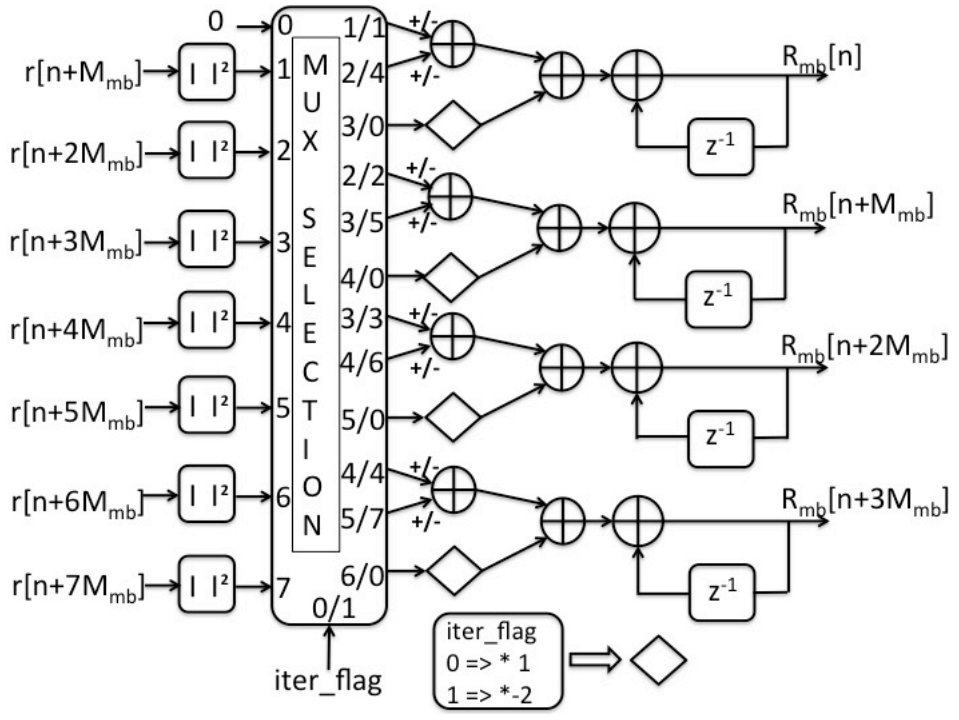

FIGURE 3.12: Proposed R = 4-Parallel PSBP Architecture for R_{mb} calculation in case of MBA. iter_flag = 0 indicates non-iterative computation mode, while iter_flag = 1 indicates iterative computation mode.

TABLE 3.5: Architectural Complexity calculation as a function of R-parallel input/output for MBA

Proposed architecture for Minn-Bhargava

Algorithm	Real Multipliers	Real Adders
P_{mb}	$4(R+3)$	$2(3R+3)$
R_{mb}	$2(R+3)$	$(3R+3)$

Kaneda's architecture for Schmidl-Cox

Algorithm	Real Multipliers	Real Adders
P_{sc}	$4R$	$2R(M_{sc}/2+1)$

3.9.3 Comparison of Architectural Complexity

Architectural Complexity as a function of R-parallel output for PSBP architecture and Kaneda's architecture in terms of number of real multipliers and adders required for realization of P_{sc}/P_{mb}. Figure 3.13 shows the number of real multipliers used. It can be observed difference of extra multipliers remains constant for all values of R. Figure 3.14 shows the number of real adders used. The gains in adders for PSBP architecture increases for higher values of R and reaches around 81% at $R = 16$.

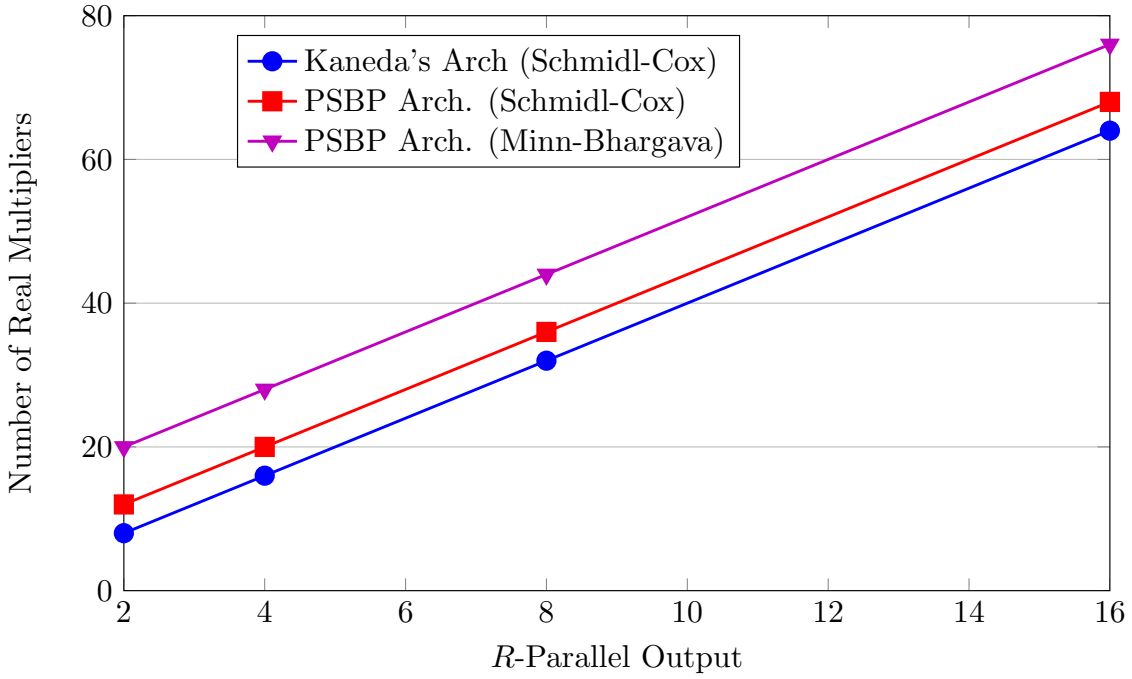

FIGURE 3.13: Multiplier requirement as a function of R-parallel output for PSBP and Kaneda's architecture, $M = 32$

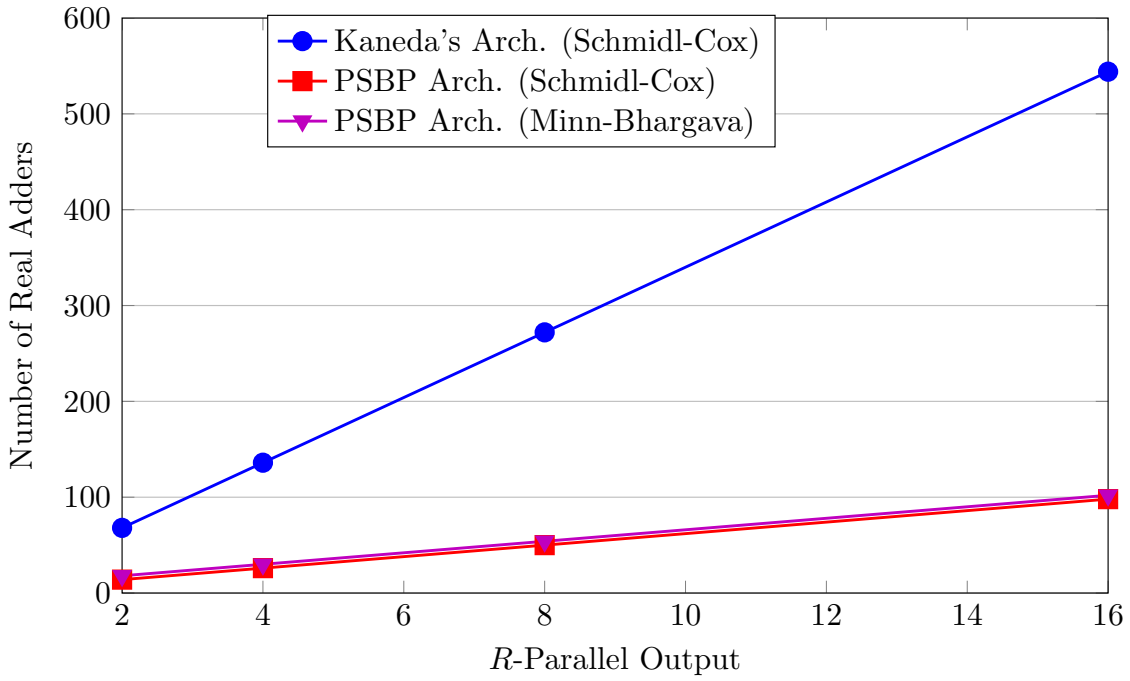

FIGURE 3.14: Adder requirement as a function of R-parallel output for PSBP and Kaneda's architecture, $M = 32$

3.10 Full-Streaming Block-Parallel (FSBP) Architecture

The PSBP architecture in Section 3.9 takes M_{sc}/M_{mb} cycles for initial point computation and this delay is encountered every time a new block computation needs to be started. This behaviour results in increase of memory for storage of OFDM symbols and hence is not completely a real-time architecture which can process multiple parallel inputs. In this section, modification of PSBP architecture is done to make the architecture fully streaming and produce M_{sc}/M_{mb} auto-correlation output of in M_{sc}/M_{mb} cycles respectively. The modification proposed is addition of separate block which can do initial point computation in parallel with iterative block so that auto-correlation output can be produced every cycle. Initial computation block computes auto-correlation output using non-iterative equation every M_{sc}/M_{mb} samples apart and it is given to iterative block periodically to start computation on next set of blocks.

3.10.1 Proposed FSBP architecture for SCA

$R = 4$-Parallel Initial Computation block for auto-correlation and energy calculation of SCA are shown in Figure 3.15 and Figure 3.16. It takes $\frac{M_{sc}}{R}$ cycles for computation of output. The increase in resources is proportional to R-parallel input. After an initial latency of M_{sc} cycles, the initial computation block feeds PSBP block R outputs continuously for

iterative computation at regular intervals. In this case, the PSBP architecture is used in iterative mode only (iter_flag=1). This makes the FSBP architecture produce outputs every cycle and therefore operate in real-time. Table 3.6 calculates the architectural complexity of P_{sc} for FSBP architecture.

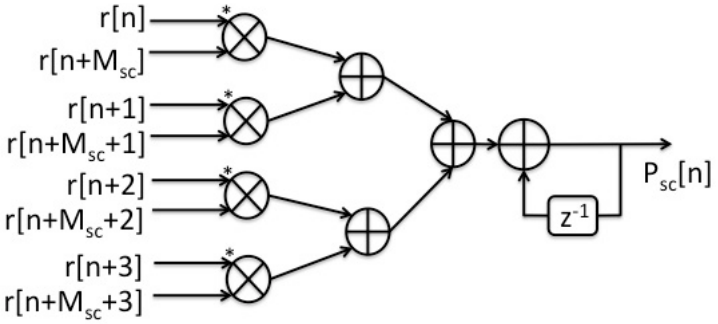

FIGURE 3.15: $R = 4$-Parallel Initial Point Auto-Correlation Computation Block for SCA

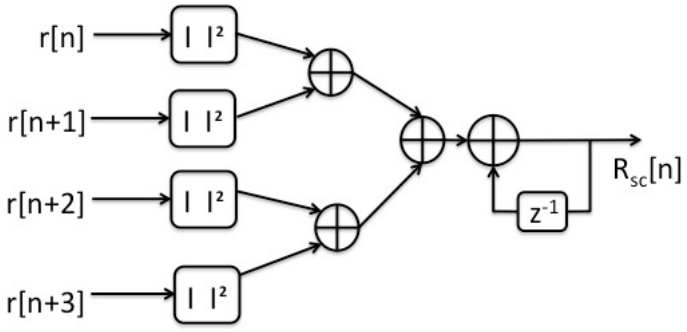

FIGURE 3.16: $R = 4$-Parallel Initial Point Energy Computation Block for SCA

TABLE 3.6: Architectural Complexity of P_{sc} for FSBP Architecture as a function of R-parallel input/output

Algorithm	Real Multipliers	Real Adders
P_{sc} (Initial Point)	$4R$	$4R$
P_{sc} (Iterative Point)	$4(R+1)$	$2(3R+1)$
P_{sc} (Total)	$8R+4$	$10R+2$
R_{sc} (Initial Point)	$2R$	$4R$
R_{sc} (Iterative Point)	$2(R+1)$	$(3R+1)$
R_{sc} (Total)	$4R+2$	$7R+1$

3.10.2 Proposed FSBP architecture for MBA

$R = 4$-Parallel Initial Computation Auto-Correlation and Energy Calculation for MBA are shown in Figure 3.17 and Figure 3.18. Table 3.7 calculates the architectural complexity of P_{mb} calculation.

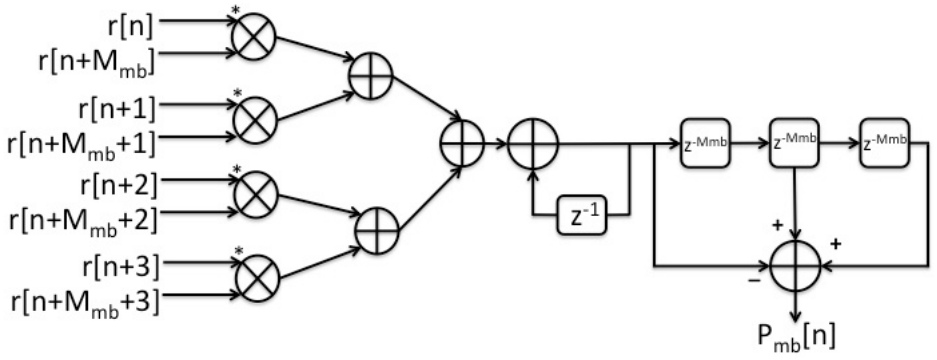

FIGURE 3.17: $R = 4$-Parallel Initial Point Auto-Correlation Computation Block for MBA

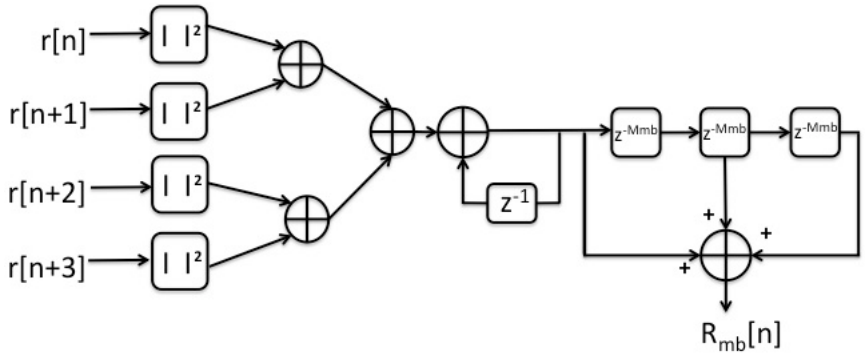

FIGURE 3.18: $R = 4$-Parallel Initial Point Energy Computation Block for MBA

TABLE 3.7: Architectural Complexity of P_{mb} for FSBP Architecture as a function of R-parallel input/output

Algorithm	Real Multipliers	Real Adders
P_{mb} (Initial Point)	$4R$	$4R$
P_{mb} (Iterative Point)	$4(R+3)$	$2(3R+3)$
P_{mb} (Total)	$8R+12$	$10R+6$
R_{mb} (Initial Point)	$2R$	$4R$
R_{mb} (Iterative Point)	$2(R+3)$	$(3R+3)$
R_{mb} (Total)	$4R+6$	$7R+3$

3.10.3 Comparison of Architectural Complexity

Comparison of Architectural Complexity of proposed FSBP architecture of SCA and MBA with Kaneda's architecture is done. Figures 3.19 and 3.20 show number of multipliers and adders for FSBP architecture as a function of R-parallel input/output respectively. Comparison based on area occupied is not straightforward since the number of multipliers required is high compared to Kaneda's architecture but the number of adders required is significantly smaller.

For comparison, area estimates of 2-stage pipelined multiplier and adder of 90 nm

technology node (Table 3.8) [50] is calculated. Area for proposed FSBP architecture and Kaneda's architecture for SCA is calculated in Table 3.9. Area savings from 21 to 74% are observed. Area for proposed FSBP architecture for MBA with Kaneda's architecture is calculated in Table 3.10. Area savings from 17 to 72% are observed.

TABLE 3.8: Area estimates for 2-Stage Pipelined Adders and Multipliers for $90nm$ technology node

Bit Width	Adder Area (μm^2)	Multiplier Area (μm^2)
4	81	612
5	109	912
8	207	2020
10	278	2953
16	530	6667
32	1235	24444
64	2588	98519

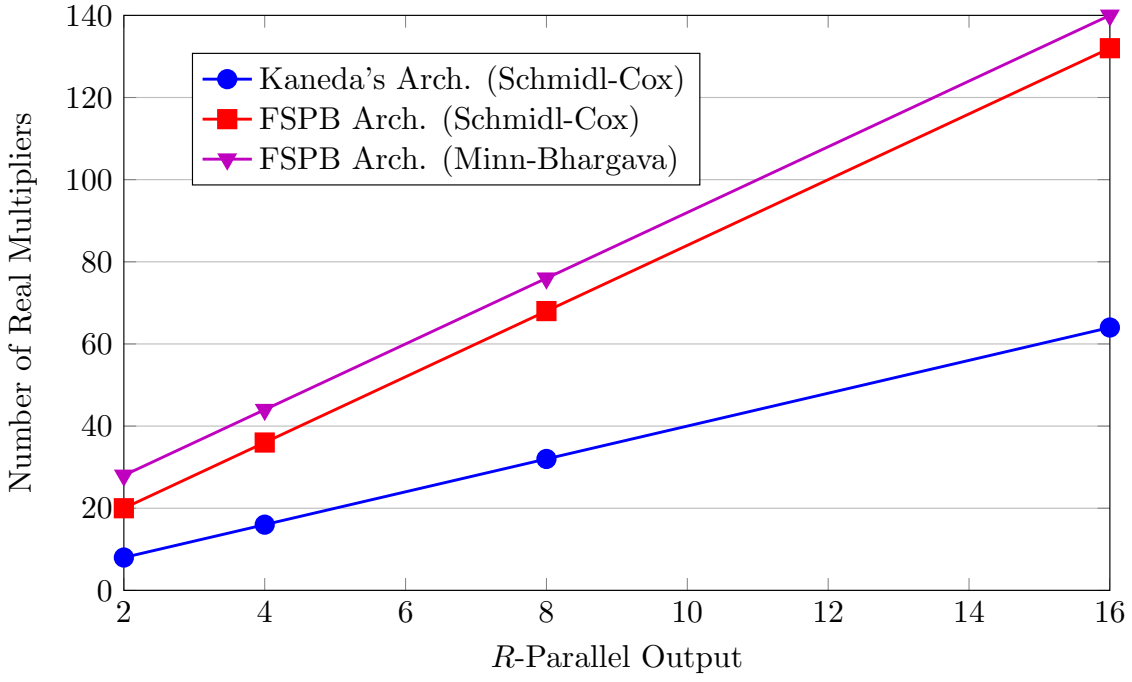

FIGURE 3.19: Multiplier requirement as a function of R-parallel output for FSBP and Kaneda's architecture, $M = 32$

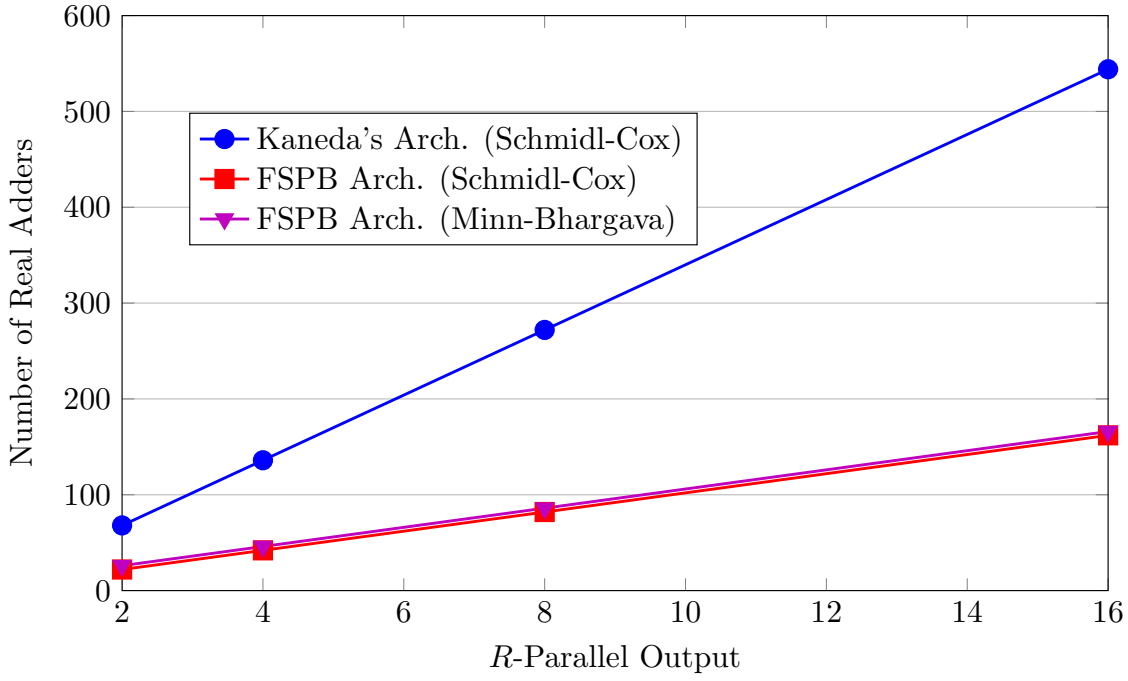

FIGURE 3.20: Adder requirement as a function of R-parallel output for FSBP and Kaneda's architecture, $M = 32$

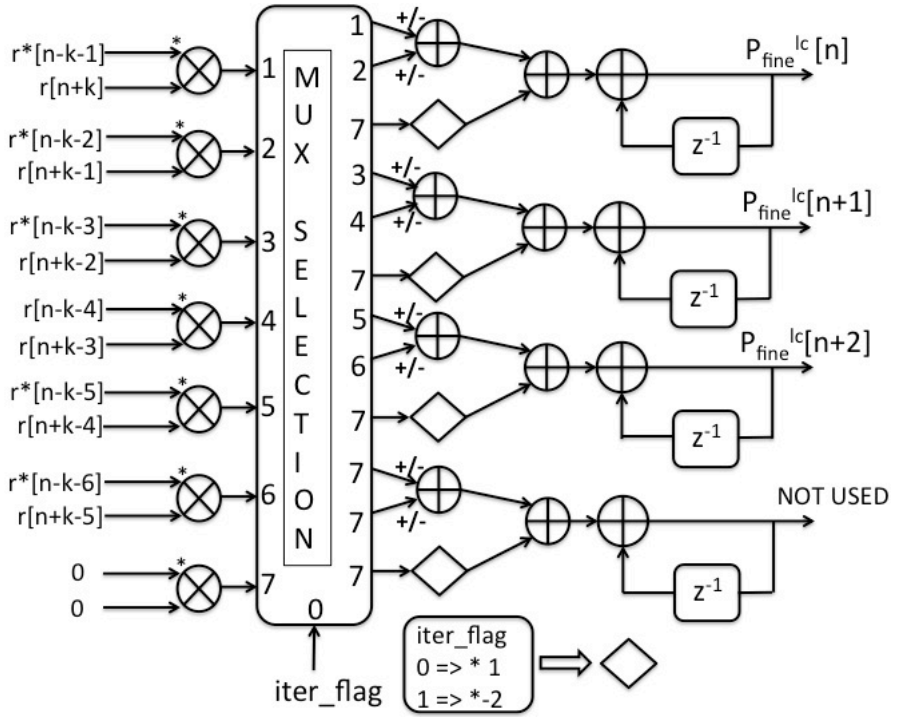

FIGURE 3.21: Parallel conjugate symmetric correlation on $R = 4$ PSPB/FSPB architecture. iter_flag $= 0$ for this operation.

TABLE 3.9: Area calculation of FSBP (Schmidl-Cox) and Kaneda's architecture at 90nm technology node for 5-bit multiplier, 10-bit adder for R = 16-Parallel input/output for Schmidl-Cox Algorithm

Training Symbol (M_{sc})	Proposed Arch. Area (μm^2)	Kaneda Arch. Area (μm^2)	Area Savings (%)
32	165420	209600	21.08
64	165420	351936	53
128	165420	636608	74.02

TABLE 3.10: Area calculation of FSBP (Minn-Bhargava) and Kaneda's architecture at 90 nm technology node for 5-bit multiplier, 10-bit adder for R = 16-Parallel input/output for Schmidl-Cox Algorithm

Training Symbol (M_{mb})	Proposed Arch. Area (μm^2)	Kaneda Arch. Area (μm^2)	Area Savings (%)
32	173828	209600	17.07
64	173828	351936	50.61
128	173828	636608	72.69

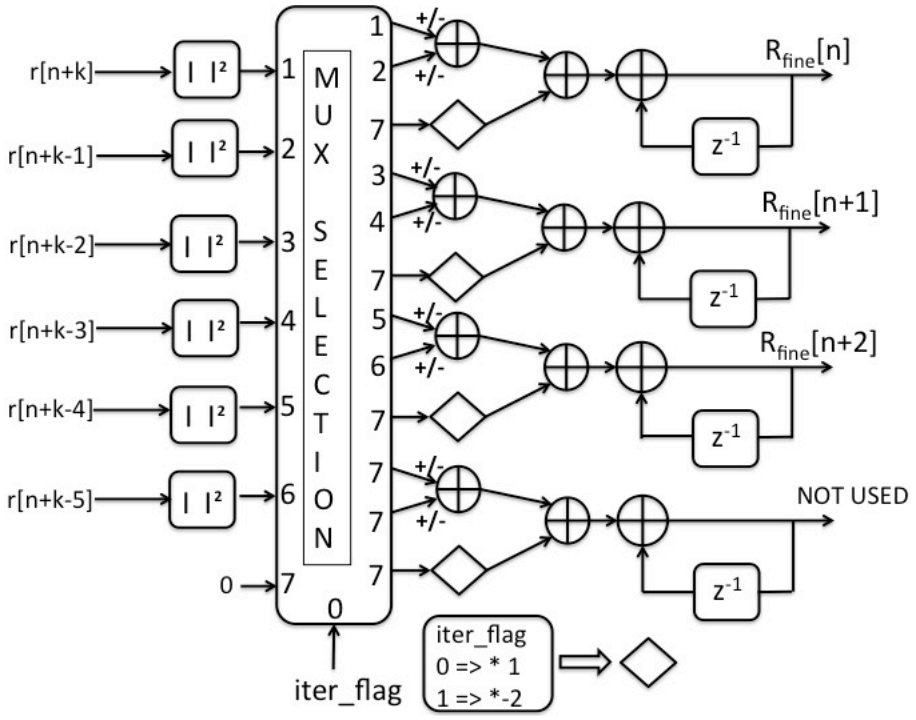

FIGURE 3.22: Parallel energy calculation on $R = 4$ PSPB/FSPB architecture. iter_flag = 0

3.11 Mapping Conjugate Symmetric Correlation onto Proposed PSPB/FSPB architecture

To support the proposed hierarchical synchronization algorithm, the architecture has to support conjugate symmetric correlation operation (Eq. 3.26, Eq. 3.27). Since proposed hierarchical algorithm uses MBA for the first step, the conjugate symmetric correlation operation is mapped onto the proposed PSPB/FSPB architecture for MBA. Figure 3.21 shows the conjugate symmetric correlation mapping on $R = 4$-Parallel PSPB/FSPB architecture. Figure 3.22 shows the mapping of energy calculation of fine time metric calculation on $R = 4$-Parallel PSPB/FSPB architecture. The mapping achieves a parallelism factor of three on $R = 4$-parallel architecture. Table 3.11 shows the factor of parallelism for conjugate symmetric correlation achieved with PSPB/FSPB architecture for different values of R-parallel auto-correlation PSPB/FSPB architecture. Note that the mapping only uses PSPB architecture and does not use initial point architecture of FSPB architecture.

TABLE 3.11: Conjugate Symmetric Correlation Parallelism Factor achieved on PSPB/FSPB architecture

R	Parallelism Factor
2	1
4	3
8	5
16	9

3.12 Conclusions

In this chapter, a low complexity time synchronization algorithm is proposed which can work in a highly dispersive channel. For complexity reduction of $\approx 80\%$, a similar MSE performance comparable to cross-correlation only estimator is observed. This algorithm is adapted to optical channel and performance of the algorithm is compared with other auto-correlation algorithms. Next, two types of block parallel architectures were proposed for synchronization algorithms. PSBP architecture provided partial streaming output and required 82% (Schmidl-Cox)/81% (Minn-Bhargava) lesser adder resources compared to Kaneda's proposal. FSBP architecture supports full streaming output and area gains of 21-72% (Schmidl-Cox) and 17-72% (Minn-Bhargava) were observed. Then, conjugate symmetric correlation operation is accelerated on the proposed PSBP/FSBP architecture for MBA to improve the timing estimation. The proposed architecture is scalable and can be generalized for use with any auto-correlation based algorithms.

Chapter 4

End-to-End Parallel Streaming Architecture for CO-OFDM System

4.1 Introduction

To reach 100 Gb/s total data rate, multi-band CO-OFDM (MB-CO-OFDM) approach is adopted to reduce the pressure on signal converters (DAC/ADC), which presently form the bottleneck in the signal processing chain. Using 50 GHz bandwidth allocated by International Telecommunication Union (ITU) standard, MB-CO-OFDM divides this total bandwidth into multiple non-overlapping sub-bands. Target data rate of 100 Gb/s requires that the total line rate to be around 117 Gb/s to accommodate for overheads in transmission. Considering this scenario, it results in the requirement of single-band to support data rates of Gb/s. Due to requirement of single-polarization to support data rates of the order of Gb/s, choice of algorithms and efficient realization of algorithms used plays a huge part in realizing the goal. OFDM frame structure choices like position and number of training symbols decides the kind of estimation algorithms. For example, training symbol based synchronization significantly reduces the complexity of detection at the receiver compared to blind synchronization. The number of training symbols used can be traded-off with the complexity of channel estimation algorithm used at the receiver, like whether to use LMS or time-frequency domain averaging techniques to keep the complexity down. Efficient realization of algorithms can be broken into two parts, namely adopting efficient parallel architectures and optimizing on fixed-point precision without incurring too much penalty on BER value.

A high level synthesis (HLS) approach has been used for realization of the CO-OFDM architecture, which is described in Section 4.2. This approach is first of its kind in case of CO-OFDM systems. Section 4.3 describes the frame structure, algorithms used for the transmitter and receiver blocks of single-polarization single-band CO-OFDM system.

Section 4.4 shows parallel architecture of the transmitter and the associated fixed-point analysis. Section 4.6 explains the parallel receiver architecture starting from frame synchronization to demapper block. Section 4.7 explains the fixed-point analysis of the receiver architecture. Section 4.8 concludes the results showing the gains due to parallel transceiver architecture and fixed-point optimizations.

4.2 A HLS Approach to Designing CO-OFDM System

CO-OFDM system realization has been done using a high-level synthesis language and the CatapultC synthesis tool [21] which allows design entry using C/C++ language with additional libraries for modeling and synthesis of FIFOs (*ac_channel*), complex data types (*ac_complex*) and fixed-point (*ac_fixed*) data types. The major attractive feature of realization using CatapultC is that same source is used for functional verification with Matlab models and simulation/synthesis of RTL code for downstream ASIC/FPGA tools. Figure 4.1 shows CatapultC synthesis flow from specification to RTL code generation and its integration with Matlab for verification of functionality and testing. The flow starts from specification about CO-OFDM system parameters like FFT size, cyclic prefix, algorithms to be used etc. Matlab is a very efficient tool for designing complex mathematical systems and offers visualization capabilities for debugging. Matlab is first used to realize the system and perform simulation varying parameters of all the blocks. The Matlab model now serves a golden reference for checking the functionality of the C/C++ implementation. Hence, integration with Matlab helps in verification of CatapultC C/C++ code.

Now, C/C++ implementation is done which implements the same blocks as in Matlab. C/C++ test bench written checks the basic functionality of the code. Extensive testing is done using Matlab integration. For Matlab testing, wrappers for interfacing with C/C++ code are generated. The wrapper and C/C++ function are imported into Matlab and compiled using MEX compiler. The generated executable is called from Matlab function and is given the same set of inputs as the golden reference. The outputs generated are compared and decision of C/C++ code satisfying the specifications are taken. Initially all data types in C/C++ code are double-precision variables. After passing initial specification test, the data types are converted to fixed-point data types. Again, the simulation of both golden reference and C/C++ is done and outputs compared to check whether it passes specifications. Here, fixed-point exploration is done to minimize the error criterion of choice. After this iteration process with Matlab, the fixed-point data types are fixed and then architecture exploration is done in CatapultC using options provided like Loop Pipelining, Loop Unrolling, FIFO sizing, mapping of memories to either register or SRAM/DRAM, etc. to meet the hardware specifications of operating frequency. Choice of interface is done based on kind of application at hand, for example, in case of streaming application *ac_channel* is used for input and output. Here the tool offers capabilities to target either ASIC or FPGA and also provides accelerated libraries for FPGA implementation. After successfully satisfying the constraints of clock frequency, the RTL code generated is simulated in

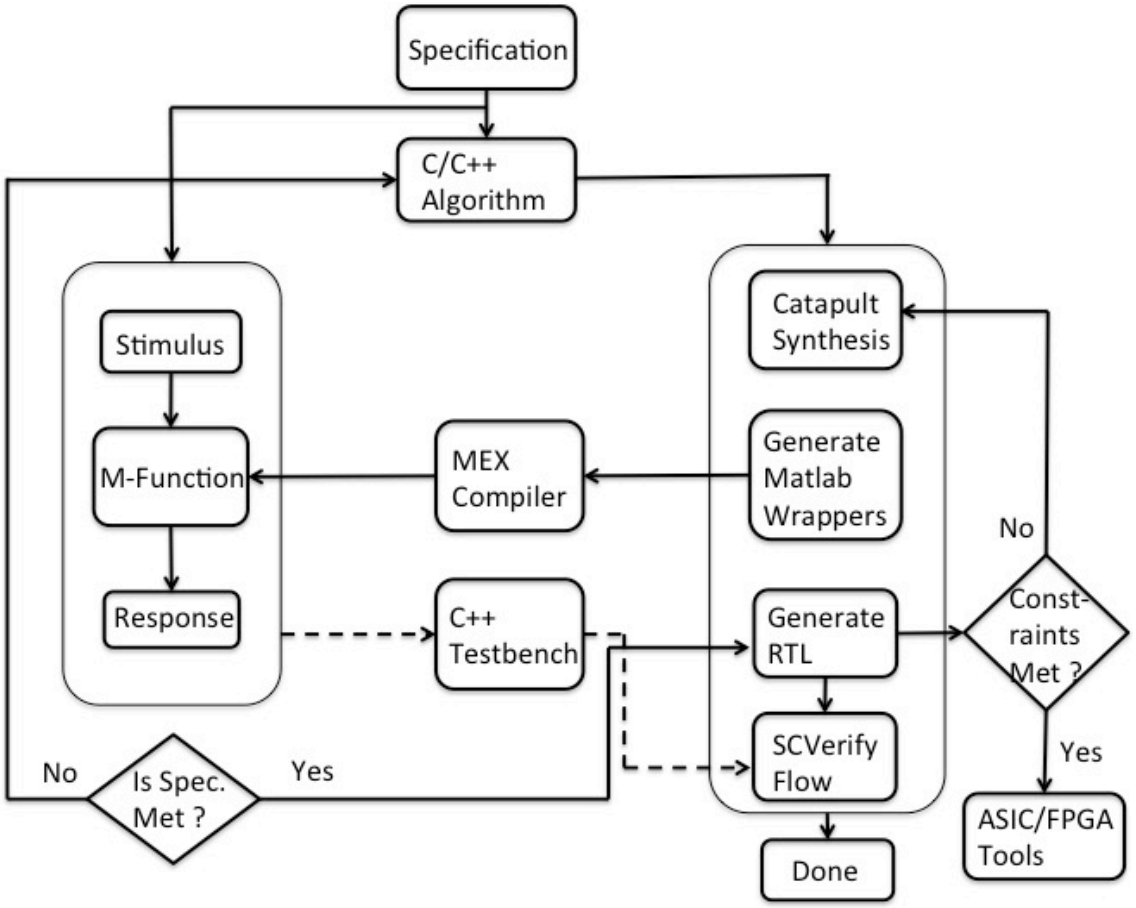

FIGURE 4.1: HLS Block Diagram of CatapultC synthesis flow and Matlab Integration

Modelsim using C/C++ test bench. This helps in verifying the generated RTL code and CatapultC also generates scripts for downstream synthesis tools.

The generated RTL code is imported into ASIC/FPGA tool flows. A RTL testbench is written which verifies the functionality of this code for testing after synthesis step. The major advantages in using CatapultC based HLS flow are

- Fixed-point exploration using C code in Matlab, which is used for RTL generation. Same fixed-point libraries used for both simulation and synthesis.
- Architecture exploration by selection of interface, loop pipelining, loop unrolling which are beneficial for DSP algorithms.

4.3 Transceiver Algorithms and Frame Structure

In this section, algorithms and frame structure selected for the realization of CO-OFDM systems are explained. The frame structure consists of training symbols (timing synchronization, fractional CFO estimation, integer CFO estimation and channel estimation) and

data symbols. Data symbols contain data and pilot symbols. Pilot symbols are used for phase offset estimation. Frame structure used for single and dual-polarization CO-OFDM systems are shown in Figures 4.2 and 4.3 respectively.

FIGURE 4.2: OFDM frame format for single polarization (Pol_X) CO-OFDM system

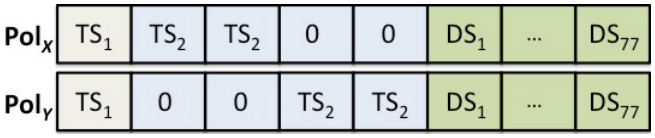

FIGURE 4.3: OFDM frame format for dual polarization (Pol_X,Pol_Y) CO-OFDM system

TS_1 (training symbol) is used for timing synchronization and fractional CFO estimation. TS_2 is used for integer CFO estimation and channel estimation. TS_2 is repeated twice [51] to improve the accuracy of channel estimation. QPSK mapping scheme is used for data and pilot symbols. In subsection 4.3.1, the selection of sizes of IFFT/FFT and cyclic prefix (CP) are discussed for a single band single-polarization CO-OFDM system and data rate achieved using this setup is calculated. Subsections 4.3.2 and 4.3.3 describe the algorithms adopted for transmitter and receiver in a single-polarization single-band CO-OFDM system.

4.3.1 Design of OFDM Parameters

According to International Telecommunication Union (ITU) standard, a total of 50 GHz is allocated for each band. Optical channel used is standard single mode fiber (SSMF), with dispersion parameters being $\eta^{CD} = 17\ \frac{ps}{nm-km}$ and $\eta^{PMD} = 12\ ps$. ADC and DAC available with effective number of bits (ENOB) of 8 bits and voltage range of 0.5 Vp-p are used. The bandwidth (B_w) used for single-band is 5 GHz, which is the sampling frequency of both DAC and ADC. This sets the bandwidth available to single-band OFDM. The maximum data rate achievable with a single band CO-OFDM is given by

$$D_b = p \cdot \log_2 M \cdot B_w \tag{4.1}$$

where D_b is the data rate in a single-band, p is the number of polarizations used, $p = 1/2$ for single/dual polarization respectively, M is given by the mapping scheme used, $M = 4$ for QPSK mapping, B_w is the bandwidth of the OFDM system. The length of cyclic prefix

is calculated by using maximum dispersion delay of SSMF channel.

$$\tau_{max} = \tau_{max}^{CD} + \tau_{max}^{PMD} \tag{4.2}$$

$$\tau_{max}^{CD} = \eta_{CD} \cdot L_f \cdot c \cdot B_w / f_0^2 \tag{4.3}$$

$$\tau_{max}^{PMD} = 3.5 \cdot \eta_{PMD} \cdot \sqrt{L_f} \tag{4.4}$$

where L_f is the fiber length in km, η^{CD} is the chromatic dispersion coefficient, c is the speed of light in m/s, B_w is the bandwidth of the system in Hz, f_0 is the LASER frequency used in Hz, η^{PMD} is the polarization mode dispersion coefficient. The length of cyclic prefix (L_{cyp}) has to be greater than maximum dispersion delay (τ_{max}). The number of samples in cyclic prefix is given by

$$N_{cyp} = L_{cyp} \cdot F_{DAC} \tag{4.5}$$

where F_{DAC} is the sampling frequency of the DAC. N_{cyp} must be sufficiently small compared to length of IFFT/FFT (N). A priori, loss in spectral efficiency is fixed

$$N = N_{cyp} \cdot \frac{1 - \epsilon_{cyp}}{\epsilon_{cyp}} \tag{4.6}$$

The constraint on maximum length of N is given by phase noise variation of LASER, which fixes the maximum value of N [52]. It limits N to values less than or equal to 256 and requires allocation of 10% of sub-carriers for phase offset tracking. Generally, a value of $\epsilon_{cyp} = 0.2$ is chosen. Table 4.1 shows one such calculation for N_{cyp} and N. The total

TABLE 4.1: Calculation of N_{cyp} and N given SMF parameters

Parameter	Value
Carrier Frequency (f_0)	193.1 THz
Sampling Frequency (F_s^{DAC})	5 GHz
Bandwidth (B_w)	5 GHz
Spectral Efficiency loss assumed (ϵ_{cyp})	20 %
Fiber Lemgth (L_f)	1000 km
Mapping scheme used (M)	QPSK
Maximum Delay (τ_{max})	0.69 ns
Cylic Prefix size (N_{cyp})	8
IFFT/FFT size (N)	256
Spectral Efficiency loss achieved	3.125 %
OFDM symbol duration (T_{sym})	52.8 ns
Sub-carrier spacing ($\frac{F_s^{DAC}}{N}$)	19.531 MHz

data rate achieved by the single-band OFDM system considering the parameters calculated and by considering loss of spectral efficiency due to forward error correction (FEC) (ϵ_{fec}), loss of spectral efficiency due to training symbols (ϵ_{tr}), loss of efficiency due to use of null

sub-carriers (ϵ_{null}),

$$D_b = (1-\epsilon_{fec})(1-\epsilon_{tr})(1-\epsilon_{null})(1-\epsilon_{cyp})\log_2(M)B_w \tag{4.7}$$

Considering $\epsilon_{fec} = 0.0627$, $\epsilon_{tr} = 0.1$, $\epsilon_{null} = 0.1$, we get $D_b = 7.362$ Gb/s for single polarization. For all further calculations, the values of $N_{cyp} = 8$ and $N = 256$ is used for designing both optical experiments and hardware implementation. To attain 100 Gb/ data rate, eight sub-bands with dual polarizations are used. The total data rate achievable is

$$D_{b,total} = 2 \cdot D_b \cdot N_{sb} \tag{4.8}$$

$$= 117.792\ Gb/s \tag{4.9}$$

where $D_{b,total}$ is the total data rate in 50 GHz channel, N_{sb} is the total number of sub-bands used. A guard band of 1 GHz is used for separating the sub-bands.

4.3.2 Transmitter Algorithm Design

In transmitter, IFFT block is the major block to design. For the transmitter, the choice of radix used for IFFT can reduce the total number of operations. The algorithmic complexity for $N = 256$ in terms of millions of operations per second (MOPS)/giga operations per second (GOPS)/tera operations per second (TOPS) is calculated for radix-2,4,2^2 and split-radix algorithms. Table 4.2 shows the number of operations per single-band for supporting 7.3 Gb/s bit rate. Then, total number of operations for supporting $D_{b,total} \geq 100$ Gb/s is shown in last column of Table 4.2.

In case of sub-band design, same IFFT is used across all sub-bands and polarizations. Any gains obtained by reduction of number of operations in single-polarization single-band IFFT is multiplied across all polarizations and sub-bands. Hence, a choice of low-complexity blocks in single-band in MB-CO-OFDM results is savings of area that will be multiplied by number of sub-bands used. From the last column of Table 4.2, it can be seen that total savings of 800 GOPS for radix-4/2^2 over radix-2 is obtained, while split-radix gains 200 GOPS over radix-4/2^2. A choice of radix-2^2 is made over radix-4/split-radix due to lesser complex parallel architecture. GOPS calculation is done as follows: GOPS = Total Operations $\cdot\ B_w/N$. TOPS = GOPS $\cdot$ 16.

TABLE 4.2: Algorithmic Complexity for calculation of N output for IFFT size of 256

Radix Used	Real Multipli-cations	Real Additions	Total Operations	GOPS (D_b) for 7.3 Gb/s	TOPS ($D_{b,total}$) for 117 Gb/s
Radix-2	4096	6144	10240	294.4	4.7
Radix-4	3072	5632	8704	248.2	3.9
Radix-2^2	3072	5632	8704	248.2	3.9
Split-Radix	2731	5462	8193	233.6	3.7

4.3.3 Receiver Algorithm Design

The algorithms used for timing synchronization, fractional CFO estimation, FFT, integer CFO estimation, channel estimation and equalization, phase error estimation and compensation are described. Algorithmic complexity for calculation of N outputs is shown and the total complexity for 117 Gb/s system is also calculated. Optimizations obtained by using specific data format of training symbols are used to eliminate multiplications. For example, in case of LS channel estimation, multiplication by reference symbol of $[\pm 1 \pm 1j]$ can be converted into a look-up table and hence complex multiplications can be avoided. Wherever such optimizations are used, reduction in complexity is calculated.

- Coarse Time Synchronization - The algorithm used is the proposed algorithm due to its superior performance over other auto-correlation algorithms. The algorithmic complexity for calculation of N outputs is shown in Table 4.3, for one sub-band and for 117 Gb/s output. Fractional CFO is estimated using auto-correlation value at the index corresponding to start point.

$$\begin{aligned} P_{mb}[n+1] = P_{mb}[n] - r^*[n] \cdot r[n + M_{mb} \\ - r^*[n + 3M_{mb} \cdot r[n + 4M_{mb}] \\ + 2 \cdot r^*[n + 2M_{mb}] \cdot r[n + 3M_{mb}] \end{aligned} \tag{4.10}$$

  where P_{mb} is the auto-correlation function, M_{mb} is the length of repeating training symbol used $[A\ A\ A\ -A]$.

TABLE 4.3: Algorithmic Complexity (auto-correlation function only) for Proposed Synchronization Algorithm

Algo. Used	Real Multipli-cations	Real Additions	Total Operations	GOPS (D_b) per sub-band	TOPS ($D_{b,total}$) for 117 Gb/s
Minn -Bhargava ($L = 4$)	3072	3072	6144	175.2	2.8

- FFT - Radix-2^2 FFT is chosen because of lower algorithmic complexity compared to radix-2 and its better architectural complexity and scalability which will be shown in Section 4.4.

- Integer CFO Estimation - Cross-correlation with known sequence of training symbol is chosen for integer CFO estimation. Consider a known complex sequence of length $N_{ifo} = N/4$, given by $z[n]$. Since, integer CFO estimation is done in frequency domain, the known sequence can be QPSK sequence with values $(\pm 1 \pm 1j)$. The

cross-correlation operation to determine the integer CFO can be written as

$$M_{ifo}[n] = |P_{ifo}[n]|^2 \tag{4.11}$$

$$P_{ifo}[n] = \sum_{m=0}^{N_{ifo}-1} r[n+m] \cdot z^*[n+m] \tag{4.12}$$

where n is the search index, $n \in [-W_s, .., -2, 0, 2, .., W_s]$, where W_s is the maximum search index. Here $W_s = 20$ is chosen as the maximum search window value. The value of N_{ifo} is chosen to be 32. The algorithmic complexity is shown in Table 4.4. Due to use of QPSK constellation $(\pm 1 \pm 1j)$, complex multiplications can be completely avoided and complexity reduced. Savings of 39.8 MOPS is obtained by this optimization.

TABLE 4.4: Algorithmic Complexity for integer CFO estimation algorithm

Algo. Used	Real Multipli-cations	Real Add-itions	Total Oper-ations	MOPS (D_b) per sub-band	MOPS ($D_{b,total}$) for 100Gb/s
IFO Estimation Non-Optimized	15360	7680	23040	3.68	59
IFO Estimation Optimized	82	2624	2706	1.3	20.8

- Channel Estimation & Equalization - The algorithms of least squares (LS) and normalized mean least squares (NLMS) algorithms have been used for channel estimation. Algorithmic Complexity is given in Table 4.5. Here, LS method's for complexity can be reduced by using multiplication by symbol $[\pm 1 \pm 1j]$ and complex multiplications avoided. This optimization works for both Single Polarization and Dual Polarization transmission. Savings of 29.2 GOPS is obtained for LS channel estimation and 21.9 GOPS is obtained for NLMS channel estimation method for single-polarization, single-band by using this optimization.

- CPE Estimation & Compensation - Pilot based CPE estimation [15] is done for estimation of LASER phase noise. Optimization can be done by using $[\pm 1 \pm 1j]$ symbol and thus complex multiplications can be avoided. Algorithmic Complexity of CPE compensation is given in Table 4.6.

From the algorithms chosen, it can be seen that savings of 800 GOPS is obtained by choosing radix-2^2 IFFT/FFT over radix-2. In the case of Integer CFO, lower complexity method cross-correlation was adopted and optimized to save 39.8 MOPS. In case of LS/NLMS channel estimation, savings of 29.2/21.9 GOPS were obtained due to optimization.

Table 4.5: Algorithmic Complexity for Channel Estimation algorithms

Algo. Used	Real Multipli-cations	Real Add-itions	Total Oper-ations	GOPS (D_b) per sub-band	TOPS ($D_{b,total}$) for 117 Gb/s
LS Single Pol. Non-Optimized	2048	768	2816	80.3	1.28
LS Single Pol. Optimized	1024	768	1792	51.1	0.81
LS Dual Pol. Non-Optimized	8192	3072	11264	321.2	5.1
LS Dual Pol. Optimized	4096	3072	7168	204.4	3.2
NLMS Single Pol. Non-Optimized	2816	2048	4864	138.7	2.2
NLMS Single Pol. Optimized	2304	1792	4096	116.8	1.8
NLMS Dual Pol. Non-Optimized	5632	4096	9728	277.4	4.4
NLMS Dual Pol. Optimized	4608	3584	8192	233.6	3.7

Table 4.6: Algorithmic Complexity for CPE Compensation

Algo. Used	Real Multipli-cations	Real Add-itions	Total Oper-ations	GOPS (D_b) per sub-band	GOPS ($D_{b,total}$) for 100Gb/s
CPE Estimation Optimized	1024	512	1536	43.8	700.8

4.4 Parallel Transmitter Architecture

In this section, parallel architecture for CO-OFDM transmitter is explained. Proposed parallel architecture of transmitter consists of a parallel mapper, IFFT and pre-emphasis block where number of parallel output times FPGA clock frequency is equal to DAC clock frequency. The mapper block supports mapping of normalized constellations from BPSK, QPSK, 16-QAM and 64-QAM. It is implemented using look-up table (LUT) where a single table is used to support all the above constellations. Since CO-OFDM Transmitter complexity depends on IFFT size [53], choice of radix plays a big role in deciding the overall complexity. The IFFT is implemented using radix-2^2 algorithm [54] which uses the same number of complex multipliers as radix-4 but uses radix-2 butterfly unit as basic block of computation, which is much simpler than radix-4 basic block of computation. There are mainly two types of pipelined architectures [54] for IFFT:

- Feed-forward
  1. Multipath Delay Commutator (MDC)
  2. Single-path Delay Commutator (SDC)
- Feedback
  1. Multipath Delay Feedback (MDF)
  2. Single path Delay Feedback (SDF)

Since feedback based architectures do not provide parallel outputs every clock cycle, only feed-forward based architectures are considered. Similar comment holds good for SDC feed-forward architecture. Only MDC feedforward architectures can provide parallel outputs every clock cycle and can be parallelized to provide higher number of parallel outputs. The IFFT equation of radix-2^2 is given by

$$x(n_1 + 2n_2 + 4n_3) = \sum_{k_3=0}^{\frac{N}{4}-1} \Big[H(n_1, n_2, k_3) \cdot W_N^{k_3(n_1+2n_2)}\Big] W_{\frac{N}{4}}^{n_3 k_3} \tag{4.13}$$

$$H(n_1, n_2, k_3) = \Big[X(k_3 + (-1)^{n_1} X(k_3 + \frac{N}{2}\Big] + (-j)^{n_1+2n_2}\Big[X(k_3 + \frac{N}{4}) + (-1)^{n_1} X(k_3 + \frac{3N}{4})\Big] \tag{4.14}$$

where $x[n]$ is the IFFT output, $X[k]$ is the input, N is the size of IFFT, W is the twiddle factor multiplication. There are two kinds of architecture based on order of input for radix-2^2 MDC IFFT. A novel architecture shown in Figure 4.4 is proposed based on input order supplied with even and odd indices [55] separated. The proposed architecture has more uniform routing architecture, but uses one extra complex multiplier compared to previously proposed architecture shown in Figure 4.5 [5]. In Figure 4.5, the inputs are applied in normal order. The routing architecture is more complicated.

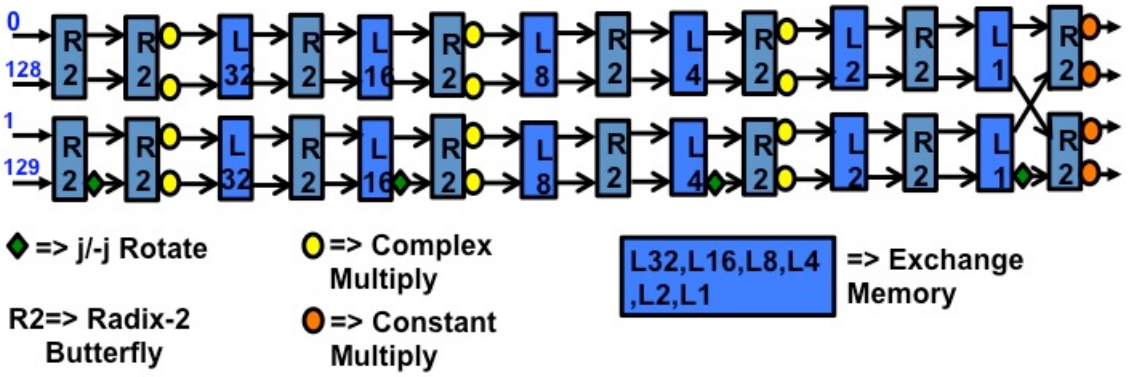

FIGURE 4.4: IFFT/FFT Architecture of 4-Parallel radix-2^2 for $N = 256$, when input is given in even and odd index order

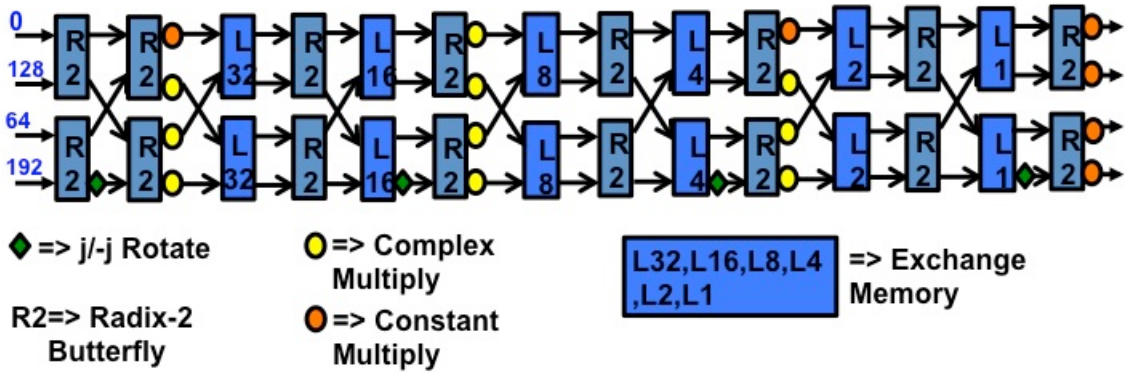

FIGURE 4.5: IFFT/FFT Architecture of 4-Parallel radix-2^2 for $N = 256$, when input is given in normal order

Since architecture of Figure 4.5 uses one less complex multiplier compared to proposed architecture (Figure 4.4), the architecture with normal input order is chosen. Table 4.7 compares the architectural complexity of radix-2^2 with radix-2/4/8/16 for different parallel outputs. It shows the scalability of radix-2^2 for different parallel outputs compared to radix-2/4/8/16. The amount of resources required by radix-2^2 is closest to the minimum resources for all number of parallel outputs. Comparatively, as number of parallel outputs increases, lower radix IFFT consumes more resources.

4.5 Fixed Point Analysis of Transmitter Architecture

Fixed-point analysis of the transmitter architecture is done in this Section. A normalized IFFT is implemented where $\frac{1}{\sqrt{N}}$ scaling is applied to both IFFT and FFT equations. After addition of cyclic prefix, the output signal is then scaled and clipped so as to match the input amplitude range of DAC. The DAC takes parallel samples from output of scaling block and converts it into analog output. The mapper receives binary data at input and maps it onto complex constellations, which is represented by fixed-point data type. The mapper outputs complex symbols from a normalized constellation of either BPSK, QPSK, 16-QAM or 64-QAM. Since 64-QAM has the largest range of output, the mapper's dynamic

TABLE 4.7: Architectural Complexity (normal input order) for full streaming outputs for $N = 256$, with input and output in natural order. Resource count is generated by using SPIRAL tool [4] for radix-2/4/8/16 and using [5] for radix-2^2

Radix	Multipliers	Adders
R = 2-Parallel Output		
2	28	46
2^2	24	44
R = 4-Parallel Output		
2	48	88
4	36	82
2^2	36	82
R = 8-Parallel Output		
2	84	172
4	72	164
8	60	162
2^2	72	164
R = 16-Parallel Output		
2	156	340
4	120	320
8	120	324
16	108	318
2^2	108	310

range is fixed by maximum and minimum values of 64-QAM output. Consider the IFFT equation implemented by the transmitter:

$$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X[k] \cdot e^{j2\pi nk/N} \tag{4.15}$$

From a fixed-point design perspective of IFFT, the main observation is that IFFT block provides input to DAC, which is precision limited to 6-8 bits. Hence, the output precision of IFFT is precision limited by precision input of DAC. This is opposite of that in receiver, where FFT occurs after ADC block. Hence, fixed-point precision computation at IFFT and FFT needs to be different and this asymmetry can be used for resource optimization in case of IFFT. Computation precision of IFFT which is closer to DAC precision is sufficient and any extra precision used for calculation in IFFT will be discarded at the DAC. Based on this observation, IFFT area optimization is done.

Table 4.8 shows the variation of Root Mean Square Error (RMSE) with resolution of input/output bits (W_i) and twiddle factor inputs (W_t). RMSE for fixed-point output is evaluated using

$$RMSE = \sqrt{\Big(\sum_{n=0}^{N-1} |S_n - T_n|^2 \Big)/N} \tag{4.16}$$

where S_n is the actual fixed-point output of IFFT while T_n is the double-precision floating-point output used as reference. Figure 4.6 shows a semilog plot of mean of RMSE as a function of variation of W_i for different values of W_t. It can be observed that minimum value of $W_t \geq 7$ is required to ensure low value of mean of RMSE. The value of W_i chosen depends on the input resolution of DAC. Next, parallel architecture of transmitter with different values of $W_t \geq 7$ and $W_i \geq 6$ is generated with CatapultC HLS tool. It allows hardware exploration in terms of pipelining, loop unrolling, etc. to achieve a high throughput architecture. The gains in area due to usage of lower fixed-point precision value to achieve a particular value of RMSE is explored. Table 4.9 shows the resources usage in terms of LUTs used as a function of different values of W_i and W_t. From Table 4.9, it can be seen that resource consumption of IFFT is a strong function of input precision (W_i) and a weak function of twiddle factor precision (W_t). Percentage increase in area from W_i = 6 to 10 bits for W_t = 8,9,10 bits is 57%,62%,66% respectively. Fixed-point optimization with respect to W_i does offer huge savings in resource usage.

TABLE 4.8: Mean (μ) and Standard Deviation (σ) of RMSE for variation of Bitwidths of inputs/outputs W_i and Twiddle Factor W_t

Bitwidth W_t	RMSE	Bitwidth (W_i)						
		4	5	6	7	8	9	10
4	μ	$5.5\text{x}10^{-2}$	$3.2\text{x}10^{-2}$	$2.1\text{x}10^{-2}$	$1.9\text{x}10^{-2}$	$1.8\text{x}10^{-2}$	$1.7\text{x}10^{-2}$	$1.7\text{x}10^{-2}$
	σ	$1.6\text{x}10^{-3}$	$7.7\text{x}10^{-4}$	$5.4\text{x}10^{-4}$	$5.4\text{x}10^{-4}$	$5.5\text{x}10^{-4}$	$5.3\text{x}10^{-4}$	$5.3\text{x}10^{-4}$
5	μ	$5.3\text{x}10^{-2}$	$2.7\text{x}10^{-2}$	$1.5\text{x}10^{-2}$	$1.0\text{x}10^{-2}$	$8.8\text{x}10^{-3}$	$8.4\text{x}10^{-3}$	$8.3\text{x}10^{-3}$
	σ	$1.6\text{x}10^{-3}$	$8.8\text{x}10^{-4}$	$4.3\text{x}10^{-4}$	$3.0\text{x}10^{-4}$	$2.5\text{x}10^{-4}$	$2.7\text{x}10^{-4}$	$2.6\text{x}10^{-4}$
6	μ	$5.2\text{x}10^{-2}$	$2.6\text{x}10^{-2}$	$1.3\text{x}10^{-2}$	$7.4\text{x}10^{-3}$	$4.8\text{x}10^{-3}$	$3.9\text{x}10^{-3}$	$3.7\text{x}10^{-3}$
	σ	$1.5\text{x}10^{-3}$	$7.4\text{x}10^{-4}$	$3.8\text{x}10^{-4}$	$2.2\text{x}10^{-4}$	$1.2\text{x}10^{-4}$	$1.2\text{x}10^{-4}$	$1.1\text{x}10^{-4}$
7	μ	$5.2\text{x}10^{-2}$	$2.6\text{x}10^{-2}$	$1.3\text{x}10^{-2}$	$7.0\text{x}10^{-3}$	$3.9\text{x}10^{-3}$	$2.8\text{x}10^{-3}$	$2.4\text{x}10^{-3}$
	σ	$1.3\text{x}10^{-3}$	$7.5\text{x}10^{-4}$	$3.9\text{x}10^{-4}$	$1.7\text{x}10^{-4}$	$9.5\text{x}10^{-5}$	$6.9\text{x}10^{-5}$	$6.3\text{x}10^{-5}$
8	μ	$5.2\text{x}10^{-2}$	$2.6\text{x}10^{-2}$	$1.3\text{x}10^{-2}$	$6.6\text{x}10^{-3}$	$3.4\text{x}10^{-3}$	$2.0\text{x}10^{-3}$	$1.4\text{x}10^{-3}$
	σ	$1.5\text{x}10^{-3}$	$6.8\text{x}10^{-4}$	$3.9\text{x}10^{-4}$	$2.0\text{x}10^{-4}$	$1.0\text{x}10^{-4}$	$5.7\text{x}10^{-5}$	$3.4\text{x}10^{-5}$
9	μ	$5.2\text{x}10^{-2}$	$2.5\text{x}10^{-2}$	$1.3\text{x}10^{-2}$	$6.5\text{x}10^{-3}$	$3.3\text{x}10^{-3}$	$1.7\text{x}10^{-3}$	$9.7\text{x}10^{-4}$
	σ	$1.4\text{x}10^{-3}$	$7.8\text{x}10^{-4}$	$3.3\text{x}10^{-4}$	$1.8\text{x}10^{-4}$	$1.0\text{x}10^{-4}$	$5.1\text{x}10^{-5}$	$2.8\text{x}10^{-5}$
10	μ	$5.2\text{x}10^{-2}$	$2.6\text{x}10^{-2}$	$1.3\text{x}10^{-2}$	$6.5\text{x}10^{-3}$	$3.2\text{x}10^{-3}$	$1.6\text{x}10^{-3}$	$8.3\text{x}10^{-4}$
	σ	$1.4\text{x}10^{-3}$	$7.6\text{x}10^{-4}$	$3.4\text{x}10^{-4}$	$1.9\text{x}10^{-4}$	$9.9\text{x}10^{-5}$	$4.4\text{x}10^{-5}$	$2.6\text{x}10^{-5}$

4.6 Parallel Receiver Architecture

In this section, parallel architecture of the receiver blocks in the chain are explained starting from time synchronization to demapper. Each of the blocks are parallelized and for all illustrations R = 4-parallel architecture is shown. Figure 4.7 shows the end-to-end connectivity of the parallel processing blocks starting from ADC to demapper. There are three memory blocks used for data to be stored temporarily while the different estimation blocks perform processing.

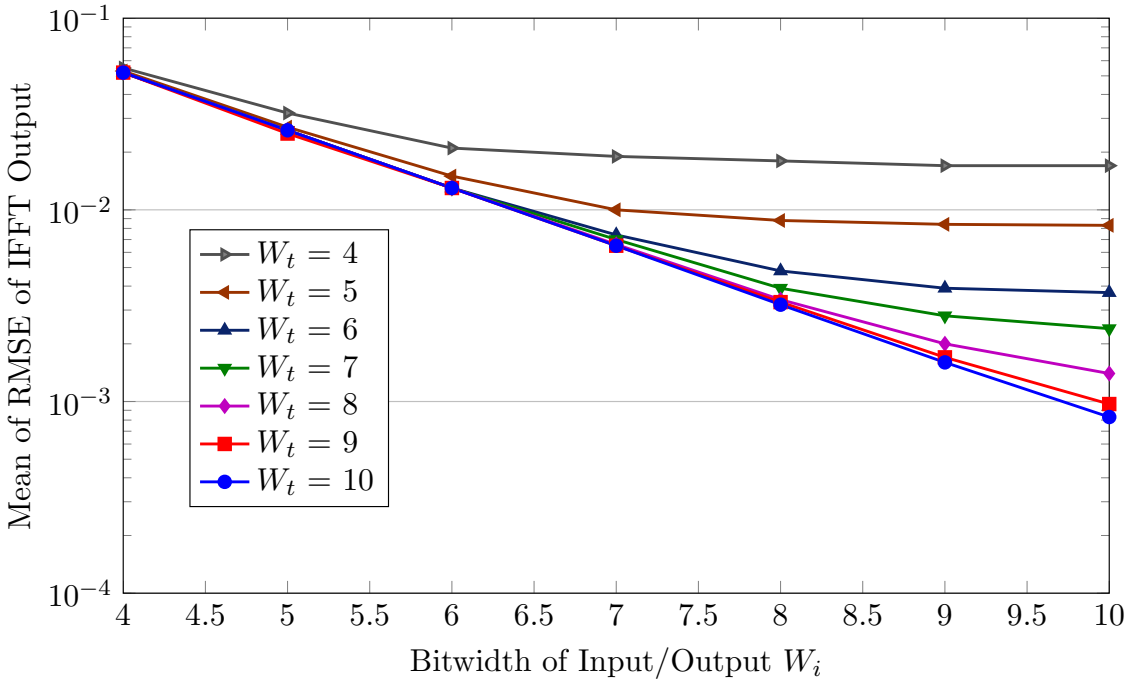

FIGURE 4.6: Plot of Mean of RMSE output of IFFT as function of W_i and W_t.

TABLE 4.9: Area Occupied for variation of Bitwidths of inputs/outputs W_i and Twiddle Factor W_t

Bitwidth W_t	Area in Xilinx FPGA	Bitwidth (W_i)		
		6	8	10
8	LUTs	38294	50187	60186
	DSP Multipliers	36	36	36
9	LUTs	38922	52543	63107
	DSP Multipliers	36	36	36
10	LUTs	40165	54652	66763
	DSP Multipliers	36	36	36

1. Time/Frequency Synchronization Memory - It receives data from ADC and gives it to the synchronization block. This memory is organized to read/write $R = 4$ samples every cycle and provides input to $R = 4$-parallel time synchronization block. Along with this memory, there is also a ping-pong memory block which also processes $R = 4$ samples every cycle. These two memory blocks together provide eight samples every cycle required for the time synchronization block. The size of synchronization memory is $6N$ samples, while the ping-pong memory is N samples. Figure 4.8 shows the organization of data in the synchronization memory for $R = 4$-Parallel input. Data organization in the ping-pong memory is the same as in this memory. After start of symbol estimation and CFO estimation, the starting address is given to FFT memory. The size and data organization of FFT memory is exactly the same

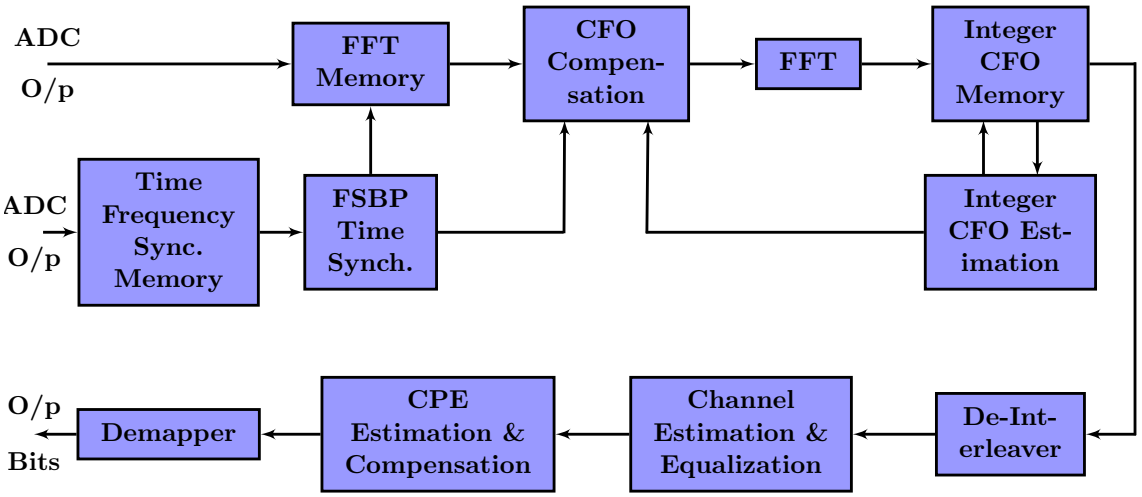

FIGURE 4.7: Proposed CO-OFDM Receiver Architecture Block Diagram

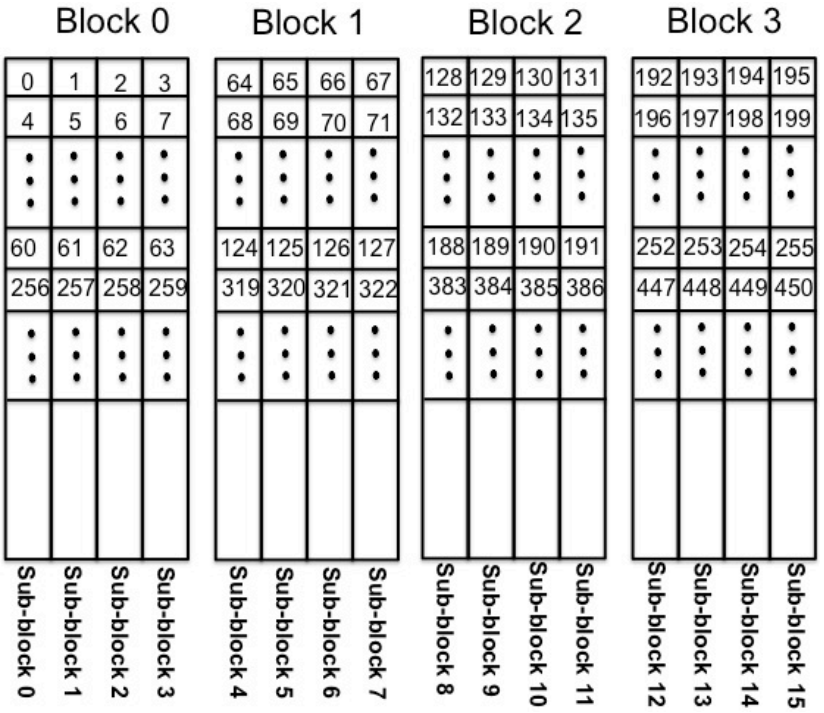

FIGURE 4.8: Data organization in the Synchronization Memory

TABLE 4.10: Architectural Complexity of Time/Frequency Architecture for $R = 4$-Parallel input/output

Algorithm	Real Multipliers	Real Adders	Total Memory Locations
P_{mb} (Total)	44	46	4096

as synchronization memory.

2. Time/Frequency Synchronization Block - It uses Full-Streaming Block-Parallel (FSBP) architecture presented in Chapter 3 to provide real-time low-complexity synchronization. Architectural complexity is calculated in Table 4.10, which also includes memory requirement of synchronization memory.

3. CFO Compensation - This block receives data from FFT memory after removal of cyclic prefix. It receives fractional CFO estimate from synchronization block and integer CFO estimate from integer CFO estimation block. It compensates the CFO by using $R = 4$-parallel multipliers. Architectural Complexity is 16 real multipliers and 16 real adders.

4. FFT - It receives CFO compensated data and outputs data in frequency domain. It uses radix-2^2 $R = 4$-parallel architecture. The architectural complexity is given in Table 4.7.

5. Integer CFO Estimation - Figure 4.9 shows the parallel architecture for integer CFO estimation using Equation 4.11. The look-up table implemented for multiplying by complex conjugate of reference input is given in Table 4.11. Hence it requires only two adders for performing multiplication with conjugate of reference input symbol. IFO estimation block uses a $4N$-size memory which stores the input till output is computed. The delay incurred is equal to receiving $3N$ amount of samples. IFO compensation involved at this point is done by reading from starting address which is equal to integer CFO predicted. Architectural complexity is given in Table 4.12.

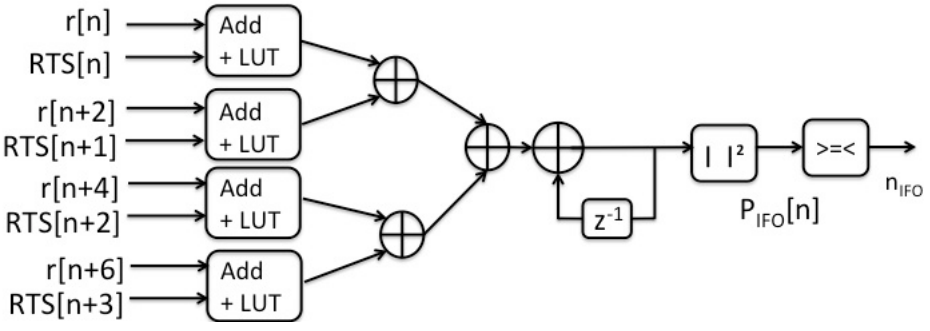

FIGURE 4.9: Parallel Architecture for IFO Estimation

TABLE 4.11: Look-up table implemented for complex multiplication of conjugate of reference symbol with input $r = a + jb$.

Reference QPSK Value	Real Output	Imaginary Output
$1 + j$	$a + b$	$-a + b$
$1 - j$	$a - b$	$a + b$
$-1 + j$	$-a + b$	$-a - b$
$-1 - j$	$-a - b$	$a - b$

TABLE 4.12: Architectural Complexity of IFO Estimation Architecture for $R = 4$-Parallel input

Algorithm	Real Multipliers	Real Adders	Memory Size
IFO Estimator	2	19	2048

6. De-interleaver - It removes the ununsed sub-carriers in the OFDM symbol and provides data and pilot sub-carriers to the next block. It also calculates energy of the non-zero sub-carrier samples. Architectural complexity is 8 real multipliers and 512 memory locations.

7. Channel Estimation & Compensation - Figure 4.10 shows the architecture of the channel estimation and equalization block for single sample input. It receives input sample and energy calculated from de-interleaver block. It also gets reference symbol for LS channel estimation or old NLMS channel estimate from memory. The updated value is written back to memory. Channel equalization is done using $H_{k,old}$ and updated H_k value calculated is written to memory for use in next iteration. The multiplexer selects input to be given to LS channel estimator or NLMS estimator. The LUT block is used to calculate the inverse of input energy. Architectural Complexity for $R = 4$-Parallel Channel Estimator and Equalizer is given in Table 4.13.

TABLE 4.13: Architectural Complexity of Channel Estimator/Equalizer for $R = 4$-Parallel input/output

Algorithm	Real Multipliers	Real Adders	Total Memory Locations
Channel Estimator and Equalizer	36	40	512

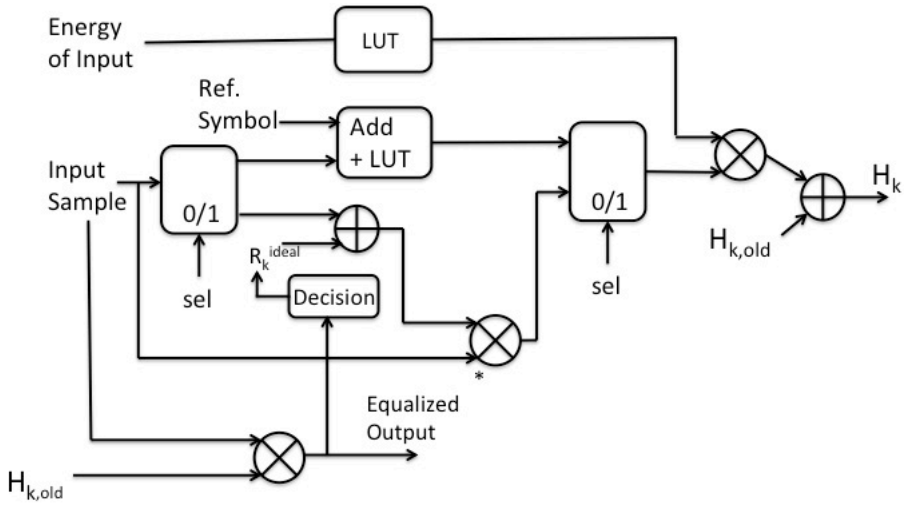

FIGURE 4.10: Channel Estimation and Equalization Architecture which supports both LS and NLMS equalizers

8. Common Phase Estimation & Compensation - Figure 4.11 shows the architecture of CPE estimation. Compensation consists of complex multiplication by using the phase error estimated. Architectural Complexity for CPE Estimation and Compensation is given in Table 4.14.

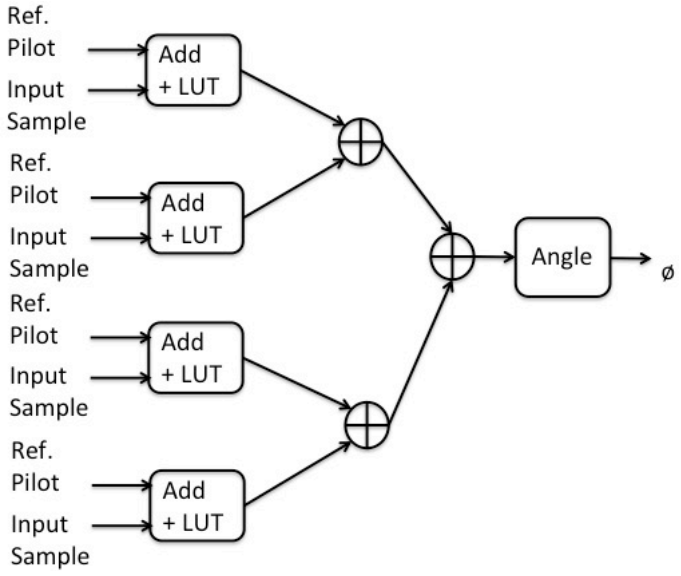

FIGURE 4.11: CPE Estimation Block

4.7 Fixed-point Analysis of Receiver Architecture

The goal of fixed-point analysis is to reduce the bitwidth used for computation in all DSP blocks of the receiver without degrading the performance significantly. The analysis gives details about the bitwidth for each of the DSP blocks in the receiver chain. It helps

TABLE 4.14: Architectural Complexity of CPE Estimator and Compensator for $R = 4$-Parallel input/output

Algorithm	Real Multipliers	Real Adders
CPE Estimator and Compensator	16	22

in identification of blocks whose precision affect more BER more significantly compared to others. This helps in aggressive optimization of such blocks. After selection of fixed-point bitwidths of all the blocks, the area vs. bitwidth variation for each of the blocks is calculated. This table helps explore optimizations which can lead to huge area savings in large blocks like FFT, channel estimator, etc with certain loss in BER. Finally, the area occupied by individual blocks after fixed-point optimization is shown in a pie-chart.

4.7.1 Analysis & Choice of Fixed-point Precision

Performance of the system using floating-point computation is considered as reference for fixed-point optimization. The conversion of the floating-point system to fixed-point system is done in a step-by-step manner. Starting from the first block of Time/Frequency synchronization, each block is converted first from floating-point computation to higher precision of fixed-point computation (generally 16-bit fixed-point number). Then, performance attained by this higher precision fixed-point computation is noted and then precision of the blocks is lowered in a linear manner. The results of this process is shown in Table 4.16 for different configurations after optimization. The various configurations used are detailed in Table 4.15. Figure 4.12 shows the performance of the fixed-point receiver for various configurations for homodyne setup. Fixed-point optimization for CFO compensation block and integer CFO estimation block is done using BER vs. OSNR curve of heterodyne setup. Table 4.18 shows the different configurations explored for heterodyne setup and Figure 4.13 shows the performance of the fixed-point receiver.

4.7.2 Area vs. Precision

Variation of area vs. fixed-point is plotted for all major blocks of the receiver. All the blocks were coded with CatapultC HLS tool. The design input is done using C language with fixed-point library (*ac_fixed*) for modeling and synthesis of fixed-point effects. It supports FIFOs (*ac_channel*) used for data exchange between blocks for streaming applications. The output of the HLS tool is Verilog/VHDL that can be targeted to either ASIC or FPGA. Architecture exploration is done using Loop Pipelining, Loop Unrolling, Mapping of memories to register array or RAMs/ROMs, Finite State Machine (FSM) coding using either gray/binary coding. Loop Pipelining has been done with Initiation Interval (II) of

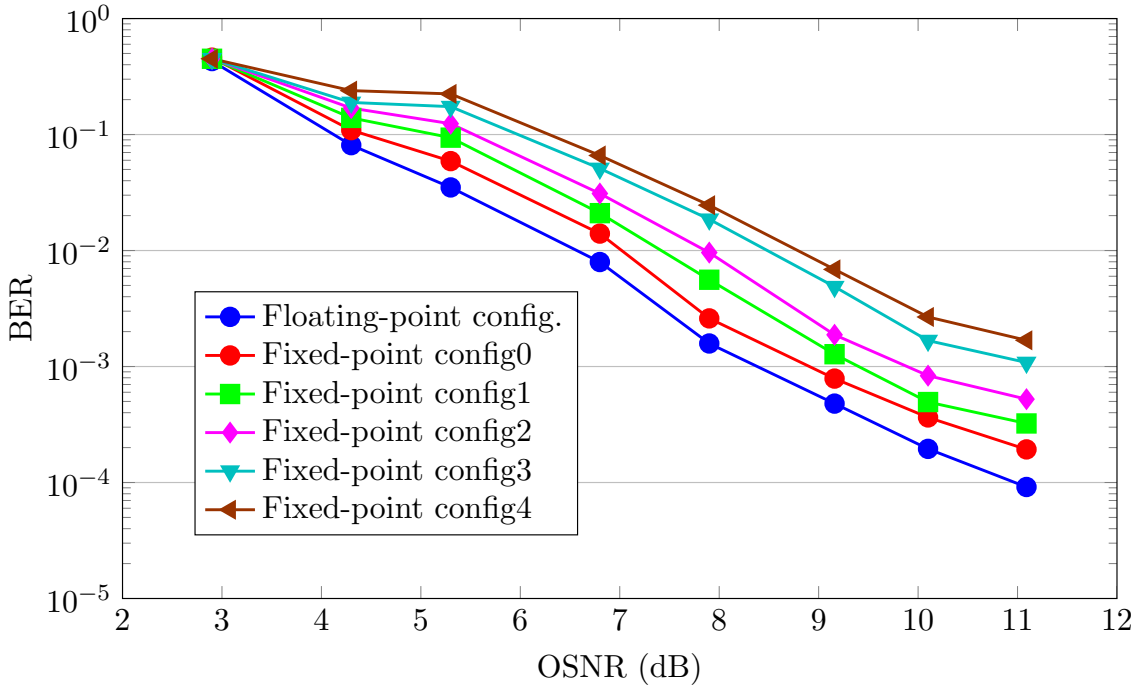

FIGURE 4.12: BER vs. OSNR plot for floating-point and various fixed-point configurations in Homodyne setup

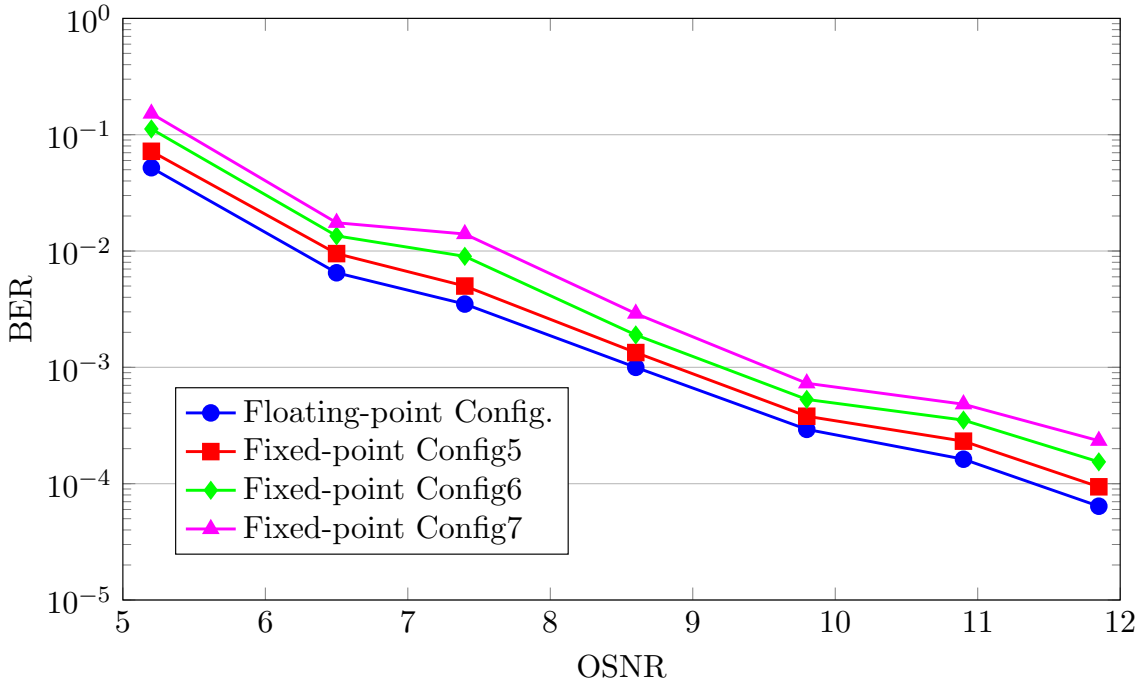

FIGURE 4.13: BER vs. OSNR plot for floating-point and various fixed-point configurations in Heterodyne setup

TABLE 4.15: Fixed-point configurations for Homodyne setup

Block Name	Fixed-point config0	Fixed-point config1	Fixed-point config2	Fixed-point config3	Fixed-point config4
	Bitwidth W_i				
Time/Frequency Synchronization	6	6	6	6	6
CFO Compensation	10	10	10	10	10
FFT	10	10	10	8	6
Intger CFO Estimation	6	6	6	6	6
De-Interleaver	10	10	10	10	10
Channel Estimation & Equalization	12	10	8	8	8
CPE Estimation & Compensation	10	10	10	10	10
Demapper	8	8	8	8	8

TABLE 4.16: BER vs. ONSR for floating-point and various fixed-point configurations in Homodyne setup

OSNR (in dB)	BER					
	5.3	6.8	7.9	9.16	10.1	11.09
Floating-point configuration	$3.5\text{x}10^{-2}$	$8.0\text{x}10^{-3}$	$1.6\text{x}10^{-3}$	$4.7\text{x}10^{-4}$	$1.9\text{x}10^{-4}$	$9.1\text{x}10^{-5}$
Fixed-point config0	$5.9\text{x}10^{-2}$	$1.4\text{x}10^{-2}$	$2.6\text{x}10^{-3}$	$7.8\text{x}10^{-4}$	$3.6\text{x}10^{-4}$	$1.9\text{x}10^{-4}$
Fixed-point config1	$9.4\text{x}10^{-2}$	$2.1\text{x}10^{-2}$	$5.6\text{x}10^{-3}$	$1.3\text{x}10^{-3}$	$4.9\text{x}10^{-4}$	$3.2\text{x}10^{-4}$
Fixed-point config2	$1.2\text{x}10^{-1}$	$3.1\text{x}10^{-2}$	$9.6\text{x}10^{-3}$	$1.9\text{x}10^{-3}$	$8.3\text{x}10^{-4}$	$5.2\text{x}10^{-4}$
Fixed-point config3	$1.7\text{x}10^{-1}$	$5.1\text{x}10^{-2}$	$1.8\text{x}10^{-2}$	$4.9\text{x}10^{-3}$	$1.7\text{x}10^{-3}$	$1.1\text{x}10^{-3}$
Fixed-point config4	$2.2\text{x}10^{-1}$	$6.6\text{x}10^{-2}$	$2.4\text{x}10^{-2}$	$6.9\text{x}10^{-3}$	$2.7\text{x}10^{-3}$	$1.7\text{x}10^{-3}$

1, which means that the receiver reads and writes data every clock cycle. The generated Verilog/VHDL code is synthesized using Xilinx ISE tool targeted towards Virtex-7 Development Board. The blocks were designed to work at a frequency of 200 MHz. Each of the blocks of the receiver were synthesized individually and resources taken at different input/output precision values are given. Tables 4.19 to 4.25 give area calculations for the blocks in the receiver starting from Time Synchronization to CPE Estimation and Compensation.

The values of precision selected for the blocks using "Fixed-point config0" are given in Table 4.26 and plotted in Figure 4.14. If instead of "Fixed-point config1" was chosen, then a savings of 13.3% in LUTs would have been obtained for a small degradation of BER.

TABLE 4.17: Fixed-point configurations for Heterodyne setup

Block Name	Fixed-point config5	Fixed-point config6	Fixed-point config7
	Bitwidth W_i		
Time/Frequency Synchronization	6	6	6
CFO Compensation	10	8	6
FFT	10	10	10
Intger CFO Estimation	6	6	6
De-Interleaver	10	10	10
Channel Estimation & Equalization	12	12	12
CPE Estimation & Compensation	10	10	10
Demapper	8	8	8

TABLE 4.18: BER vs. ONSR for floating-point and various fixed-point configurations in Heterodyne setup

OSNR (in dB)	BER					
	6.5	7.4	8.6	9.8	10.9	11.85
Floating-point configuration	$6.5\text{x}10^{-3}$	$3.5\text{x}10^{-3}$	$1.0\text{x}10^{-3}$	$2.9\text{x}10^{-4}$	$1.6\text{x}10^{-4}$	$6.4\text{x}10^{-5}$
Fixed-point config5	$9.5\text{x}10^{-3}$	$5.0\text{x}10^{-3}$	$1.3\text{x}10^{-3}$	$3.8\text{x}10^{-4}$	$2.3\text{x}10^{-4}$	$9.4\text{x}10^{-5}$
Fixed-point config6	$1.3\text{x}10^{-2}$	$9.0\text{x}10^{-3}$	$1.9\text{x}10^{-3}$	$5.3\text{x}10^{-4}$	$3.5\text{x}10^{-4}$	$1.5\text{x}10^{-4}$
Fixed-point config7	$1.7\text{x}10^{-2}$	$1.4\text{x}10^{-2}$	$2.9\text{x}10^{-3}$	$7.3\text{x}10^{-4}$	$4.8\text{x}10^{-4}$	$2.3\text{x}10^{-4}$

TABLE 4.19: Area Occupied vs. Bitwidth for Time/Frequency Synchronization block

Time/Frequency Synchronization		
Bitwidth W_i	Area in Xilinx FPGA	Area Numbers
6	LUTs	36210
	DSP Multipliers	36
8	LUTs	44873
	DSP Multipliers	36
10	LUTs	51374
	DSP Multipliers	36

Table 4.20: Area vs. Bitwidth for CFO Compensation block

CFO Compensation		
Bitwidth W_i	Area in Xilinx FPGA	Area Numbers
6	LUTs	1807
	DSP Multipliers	15
8	LUTs	1985
	DSP Multipliers	15
10	LUTs	2437
	DSP Multipliers	15

Table 4.21: Area vs. Bitwidth for FFT block

FFT		
Bitwidth W_i	Area in Xilinx FPGA	Area Numbers
8	LUTs	73529
	DSP Multipliers	36
10	LUTs	85653
	DSP Multipliers	36
12	LUTs	97979
	DSP Multipliers	36

Table 4.22: Area vs. Bitwidth for Integer CFO Estimation block

Integer CFO Estimation		
Bitwidth W_i	Area in Xilinx FPGA	Area Numbers
6	LUTs	5987
	DSP Multipliers	2
8	LUTs	8200
	DSP Multipliers	2
10	LUTs	10843
	DSP Multipliers	2
12	LUTs	12984
	DSP Multipliers	2

Table 4.23: Area vs. Bitwidth for De-interleaver block

De-Interleaver		
Bitwidth W_i	Area in Xilinx FPGA	Area Numbers
8	LUTs	9622
	DSP Multipliers	8
10	LUTs	11543
	DSP Multipliers	8
12	LUTs	13678
	DSP Multipliers	8

Table 4.24: Area vs. Bitwidth for Channel Estimation & Equalization

Channel Estimation & Equalization		
Bitwidth W_i	Area in Xilinx FPGA	Area Numbers
8	LUTs	34520
	DSP Multipliers	36
10	LUTs	41975
	DSP Multipliers	36
12	LUTs	48467
	DSP Multipliers	36

Table 4.25: Area vs. Bitwidth for CPE Estimation & Compensation

CPE Estimation & Compensation		
Bitwidth W_i	Area in Xilinx FPGA	Area Numbers
8	LUTs	1540
	DSP Multipliers	16
10	LUTs	1756
	DSP Multipliers	16
12	LUTs	1943
	DSP Multipliers	16

4.8 Conclusions

In this Chapter, an end-to-end fully streaming parallel architecture for CO-OFDM system was proposed. For the transmitter, low-complexity radix-2^2 IFFT algorithm was used, for which scalable parallel architecture was utilized. Utilizing the idea that IFFT is present before DAC and hence limited by its precision, architecture was designed to use precision closer to DAC precision. This helps in area savings compared to FFT at the receiver which is not precision limited since it receives data from ADC. The frame structure and algorithms chosen is done to reduce long feedback loops. The proposed architecture has

TABLE 4.26: Fixed-Point Allocation and Area for all blocks of the $R = 4$-Parallel Receiver

Block Name	Bitwidth W_i	Area in Xilinx FPGA	Area Numbers
Time/Frequency Synchronization	6	LUTs	36210
		DSP Multipliers	36
CFO Compensation	10	LUTs	2437
		DSP Multipliers	15
FFT	10	LUTs	85653
		DSP Multipliers	36
Integer CFO Estimation	6	LUTs	5987
		DSP Multipliers	2
De-Interleaver	10	LUTs	11543
		DSP Multipliers	8
Channel Estimation & Equalization	12	LUTs	41975
		DSP Multipliers	36
CPE Estimation & Compensation	10	LUTs	1756
		DSP Multipliers	16
Demapper	8	LUTs	640
		DSP Multipliers	0

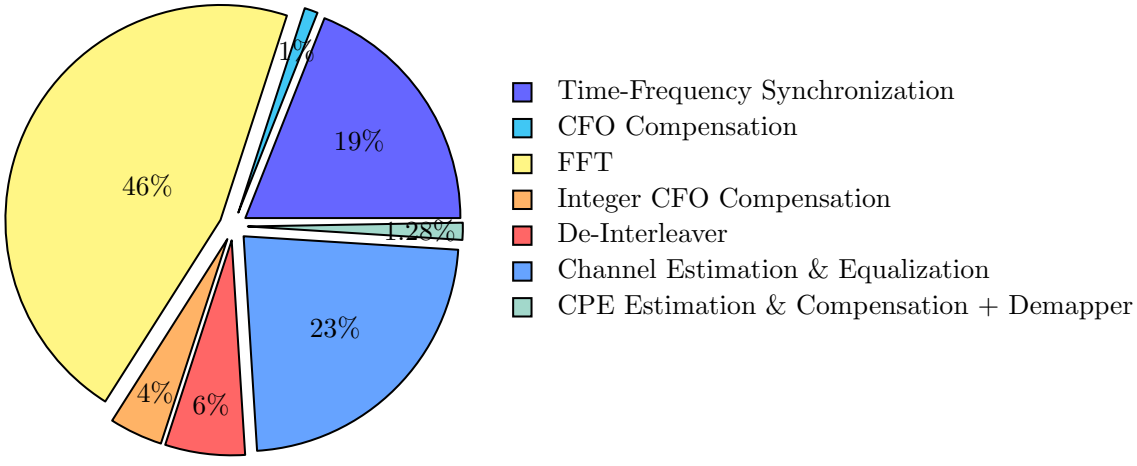

FIGURE 4.14: Pie Chart of area Occupation of all blocks of $R = 4$-Parallel CO-OFDM Receiver (Fixed-point config0)

only one feedback loop whose output is required once every 80 OFDM symbols. At the receiver, scalable parallel architecture for Time Synchronization was used. Low-Complexity Parallel blocks for Integer CFO Estimation, Channel Estimation and CPE Estimation was proposed. The implementation was done for $R = 4$-Parallel CO-OFDM Transceiver on Xilinx FPGA using CatapultC. Fixed-point exploration was done using *ac_fixed* models and whole parallel CO-OFDM receiver fits in a single Virtex-7 development board. Area vs. BER trade-off exploration was indicated.

Chapter 5

Experimental Validation of CO-OFDM System

5.1 Introduction

In this chapter, offline and real-time experiments conducted for validation of OFDM frame format, transceiver algorithms adopted are explained. In the first half, optical experiments conducted using arbitrary waveform generator (AWG) as transmitter, digital storage oscilloscope (DSO) for sampling received signal and Matlab for generating and decoding data are described. The OFDM frame format and transceiver algorithms used are from Section 4.3. The algorithm used for selection of best sampling point when using oversampling at the receiver is described in Section 5.2. The experiments are done in a step-by-step manner starting from electrical back-to-back (B2B) experiment, which is detailed in Section 5.3. In Section 5.4, the effect of addition of RF driver to the setup is explored. Section 5.5 describes optical B2B configuration using same LASER for both transmitter and receiver. Performance is characterized by plotting BER as a function of optical signal-to-noise ratio (OSNR). In section 5.6, performance of the system is explored in the presence of separate LASER sources for transmitter and receiver, which resembles real-world data transfer using optical communication systems.

The second half details experiments conducted on real-time FPGA platform. Section 5.7 provides details about the transmitter and receiver FPGA prototype platform, OFDM frame structure and algorithms used. It primarily validates the use of real-time full-streaming block-parallel (FSBP) architecture proposed for timing synchronization. The timing synchronization block provides input samples to FFT in a continuous manner. The FPGA transceiver prototype platform was developed as part of FUI 100GFLEX project. Section 5.8 gives the results achieved by the use of proposed architecture in presence of synchronous and asynchronous sampling at the receiver. Section 5.10 concludes the chapter.

5.2 Sampling Clock Offset (SCO) Estimation Algorithm

The OFDM frame format used is the same as the one proposed for developing parallel transceiver architecture. Figure 5.1 shows the frame structure used for optical experiments. consists of one initial training symbol (TS_1) which is used for time synchronization and fractional CFO estimation and of two repeated training symbols (TS_2) for integer CFO and channel estimation. A total of 77 data symbols follow these three training symbols. A single frame contains totally 80 OFDM symbols. Each data symbol contains eight pilot sub-carriers for phase estimation.

FIGURE 5.1: OFDM frame format for single polarization (Pol_X) CO-OFDM system

The algorithms used in receiver are:

- Time Frequency Synchronization - Proposed Hierarchical algorithm.
- Integer CFO Estimation - Cross-correlation with known training symbol.
- Channel Estimation - LS for initial estimation and NLMS Equalizer for tracking.
- Phase Tracking - Common Phase Error (CPE) estimation using pilots in data symbols.

In the optical experiments conducted, the signal was oversampled at the receiver and for decoding, the best sampling point needs to be found. To find the optimal sampling point in the oversampled signal, fine timing estimation algorithm was used. Fine timing estimation algorithm calculates residual timing offset in the OFDM signal after the FFT operation. The output of this operation has integer and fractional parts. The integer part gives residual integer timing offset estimate, while the fractional part gives sampling clock offset (SCO) estimate.

$$\eta_{fine} = \eta_i + \eta_{SCO} \tag{5.1}$$

where η_{fine} is the total fine timing offset estimate, η_i is the integer timing offset estimate and η_{SCO} (the fractional part) is the SCO estimate. For the estimation of fine timing offset, the algorithm proposed by Lee et. al [56] is used. It uses the phase information present in the sub-carriers to derive estimates of timing offset. Consider the received signal after FFT affected by residual timing offset after coarse timing synchronization

$$Y_j[k] = X_j[k] \cdot e^{j2\pi k(\eta_i+\eta_{SCO})/N} + N_j[k] \tag{5.2}$$

where $Y_j[k]$ and $X_j[k]$ is the k^{th} sub-carrier of j^{th} OFDM symbol, η_i is the integer timing offset, η_{SCO} is the SCO, $N_j[k]$ is the noise at the sub-carrier. Data symbols are mapped using QPSK scheme. The effect of timing offset in the frequency domain is the rotation

of sub-carrier proportional to sub-carrier number. The basic idea is subtracting phase difference between neighbouring sub-carriers and taking mean of the phase difference. An improved method is averaging phase difference calculated on a group of sub-carriers. The algorithm is given as follows

$$Yp4[j] = \frac{1}{B} \sum_{k=-B/2}^{B/2-1} Y^4[j+k], |j| < \frac{N}{2} \tag{5.3}$$

where B is the group of sub-carriers, $Yp4$ is the fourth power of group of sub-carriers of Y. Fourth power is taken to neutralize the effect of QPSK modulation on sub-carriers. The groups are selected to be continuous set of sub-carriers. The group is selected so as to exclude group of null sub-carriers. Here $B = 4$ is used for group size. The mean complex phase rotation of array of $Yp4$ is given by

$$G = \frac{1}{K} \sum_{k=0}^{K-1} Yp4[k] \cdot Yp4^*[k+1] \tag{5.4}$$

where G is the mean of complex phase rotation, it contains both integer and fractional timing offset contribution, K is the total number of group of sub-carriers. The fine symbol timing estimation is obtained by taking angle of G:

$$\eta = \frac{N}{2\pi B} \angle G \tag{5.5}$$

where η denotes total fine timing offset, $\eta = \eta_i + \eta_{SCO}$. In an oversampled signal, this calculation is done on all the sample streams obtained by down sampling. For example, in case of oversampling by a factor of 10, η_{SCO} is calculated for all 10 sample streams and sample stream with the lowest η_{SCO} is selected for further demodulation. But, calculation of η_{SCO} for all streams for every OFDM symbol in each frame results in too much computational complexity. To reduce the computational complexity, average SCO value is calculated using a 20 OFDM data symbols for single downsampled stream. Then, the next downsampled stream is chosen for calculation of SCO over next 20 OFDM symbols. It is done till all the downsampled streams are covered. Stream corresponding to minimum of the average SCO values calculated is selected for further data decoding. For example, if received signal is oversampled by 10, then for each of the 10 streams, SCO is calculated using 20 data OFDM symbols. After this calculation, stream corresponding to minimum SCO value is chosen. This method reduces the computational complexity and assuming ADC sampling clock remains stable over many OFDM frames, this calculation needs to be repeated after many OFDM frames (1000 OFDM Frames).

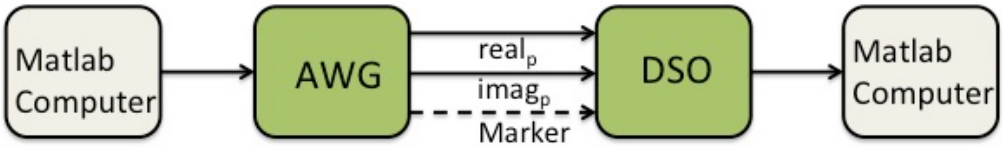

FIGURE 5.2: Configuration of Electrical B2B Experiment. Green blocks indicate analogue blocks.

5.3 Electrical Back-to-Back (B2B) Experiment

Figure 5.2 shows the electrical B2B configuration, with Matlab being used for generation and decoding data offline. The configuration consists of arbitrary waveform generator (AWG) directly connected to digital storage oscilloscope (DSO). The trigger signal for DSO is provided by the Marker signal from AWG. The objective is to characterize the Matlab transceiver in presence of DAC/ADC's limited resolution, frequency response and bandwidth of the analogue components in the configuration. Parameters of OFDM system, DAC and ADC are given in Table 5.1.

TABLE 5.1: Parameters of the Electrical B2B Experiment

Parameter	Value
N	256
N_{cyp}	8
F_s^{DAC}	1 GHz
DAC Resolution	8 $bits$
DAC Voltage Range	1 $Vp-p$
LASER Wavelength	1557.07 nm
F_s^{ADC}	10 GHz
ADC Resolution	8 $bits$
ADC Voltage Range	0.25 $Vp-p$
Number of Used sub-carriers	176
Number of Pilot sub-carriers	8
Mapping Scheme	QPSK

The spectrum of OFDM signal after AWG taken using spectrum analyzer is given in Figure 5.3. Due to high-frequency sub-carriers (sub-carriers near $N/2$) being switched off, aliasing effects are reduced. The selection of best sampling stream is done using the estimated values of η_{SCO} (shown in Figure 5.4) and it's gradient (Figure 5.5). The minimum value of absolute value of product of η_{SCO} and gradient of η_{SCO} is used as a metric for deciding the best sampling stream among the possible choices.

Theoretical BER for QPSK modulation is given by

$$BER = 0.5\ erfc\left(\sqrt{\frac{E_s}{2N_0}}\right) \tag{5.6}$$

$$erfc(x) = \frac{2}{\sqrt{\pi}}\int_x^{\infty} e^{-t^2}dt \tag{5.7}$$

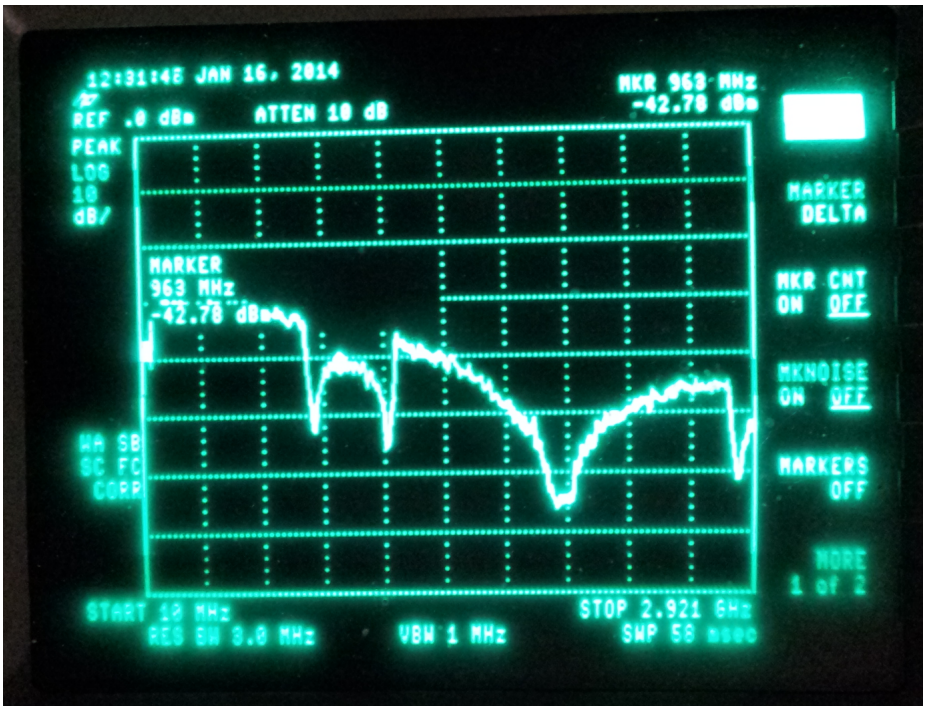

FIGURE 5.3: OFDM Signal Spectrum

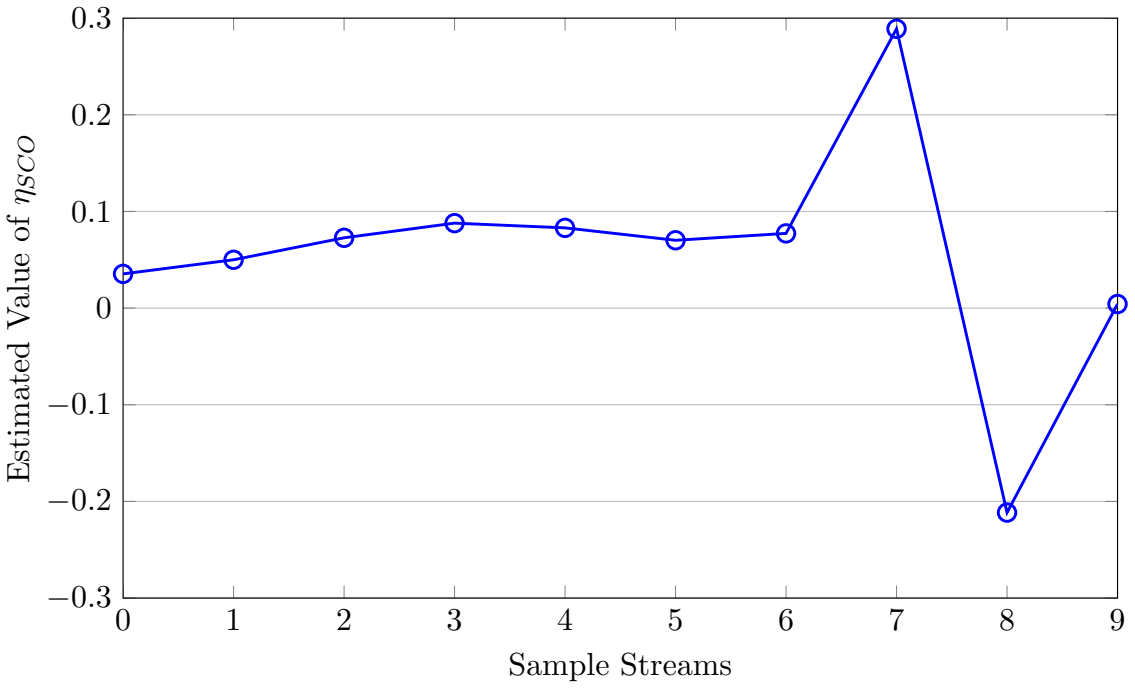

FIGURE 5.4: Estimated Values of η_{SCO}

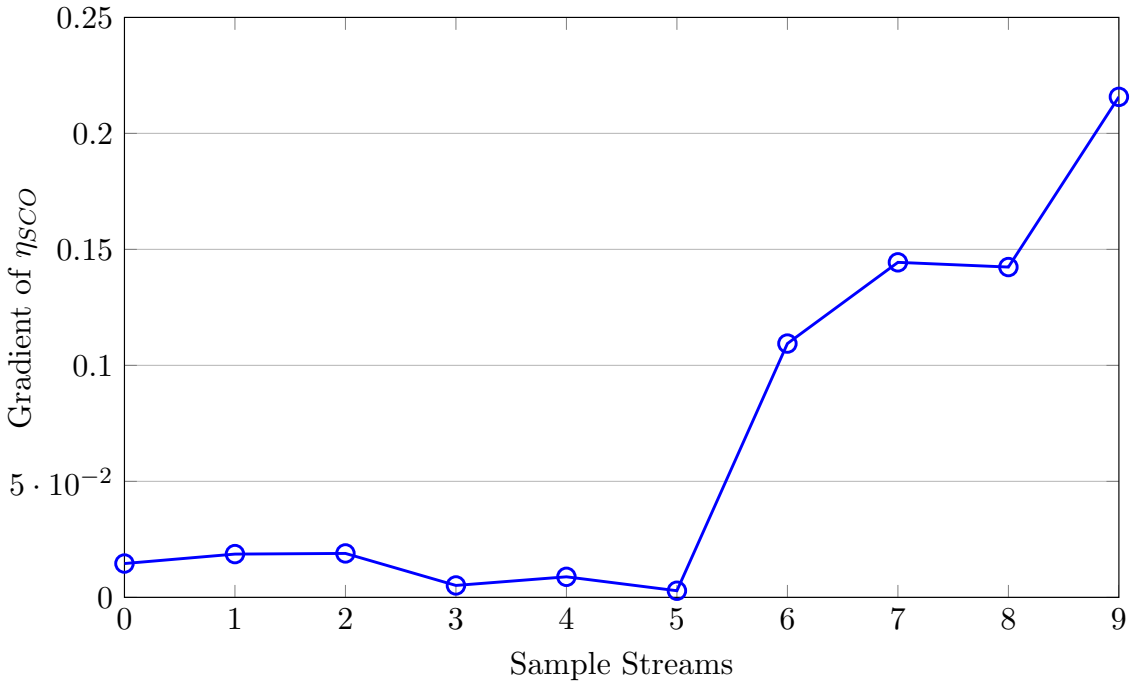

FIGURE 5.5: Gradient of estimated value of η_{SCO}

Figure 5.6 shows BER as a function of E_s/N_0. E_s/N_0 is varied by adding gaussian noise at the receiver. BER is calculated by averaging over 1000 frames in a single acquisition. Each frame consists of 77 OFDM data symbols. It can be seen that experimental BER curve follows theoretical curve very closely and validates the Electrical B2B configuration.

5.4 Electrical B2B Configuration with RF Amplifier

To generate an Optical OFDM signal, the output signal level of AWG is not sufficient to drive the optical Mach-Zender Modulator (MZM). To boost the signal, RF driver is used and its impact on the performance of CO-OFDM transceiver is evaluated. Figure 5.7 shows the electrical B2B experiment after the addition of RF driver for amplification. The output of AWG is fed into RF driver introduced. The RF driver now drives the input of DSO. A 20 dB attenuator is connected to the output of RF driver to limit the maximum voltage input to DSO. The RF driver provides a fixed gain over a large bandwidth. The parameter to vary is the peak-to-peak voltage output from AWG to input of RF driver. The range of AWG output voltage is 0.5 Vp-p to 1 Vp-p. The AWG output voltage is varied to find the value which does not saturate the RF amplifier and the value is found to be 0.6 Vp-p. The output of RF amplifier corresponding to 0.6 Vp-p after amplification by RF driver is around 10 dBm. The output spectrum after RF driver is similar to Figure 5.3. Figure 5.8 shows the variation of BER as a function of E_s/N_0. BER is calculated using average over 1000 OFDM frames and plotted in Figure 5.8 along with electrical B2B

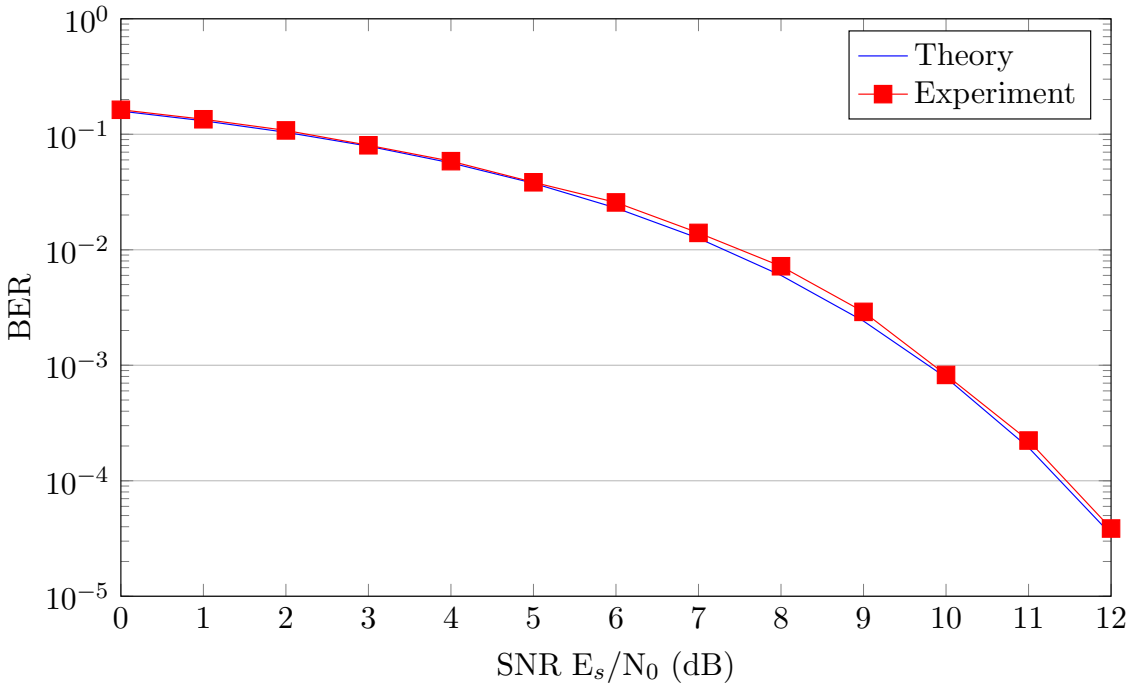

FIGURE 5.6: BER vs SNR for Electrical B2B experiment (Theoretical and Experimental)

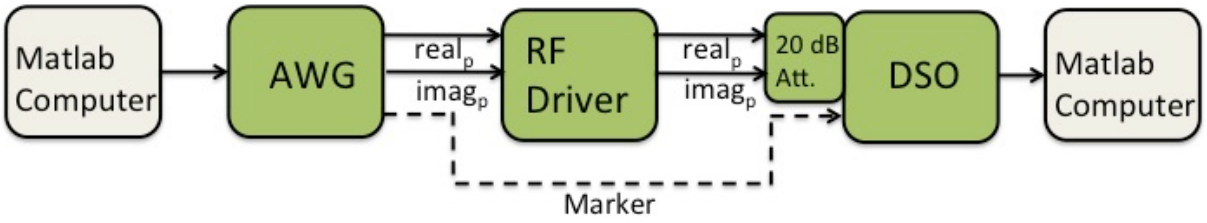

FIGURE 5.7: Configuration of Electrical B2B Experiment with RF Driver. Green blocks indicate analogue blocks.

curves. From Figure 5.8, it can be observed that since BER curve after RF driver is very close to BER curve of Electrical B2B experiment. It shows that the addition of RF driver does not introduce non-linearities in the transmission chain.

5.5 Optical B2B Configuration with Homodyne Coherent Detection

After validation of Electrical B2B configuration, optical components are introduced. Figure 5.9 shows the optical B2B configuration after the addition of optical modulator and demodulator. Since a common LASER is used for modulating and demodulating optical signal, it is called homodyne detection. The configuration is described in detail below.

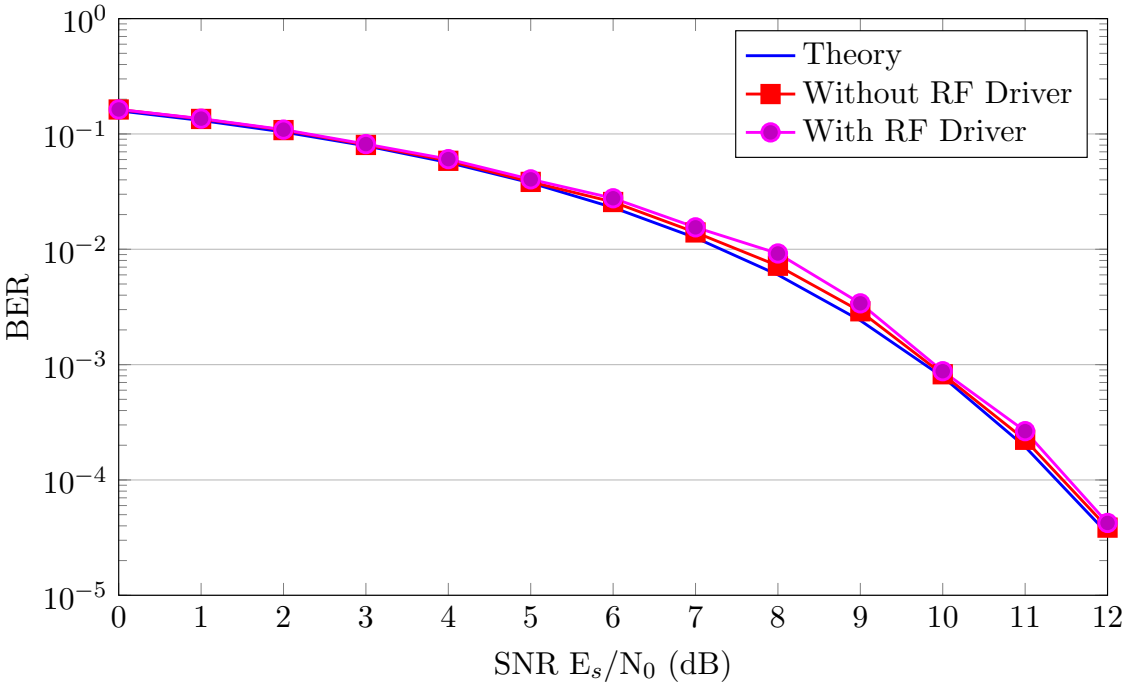

FIGURE 5.8: BER vs SNR for Electrical B2B experiment with RF driver (Theoretical, Experimental with and without RF driver)

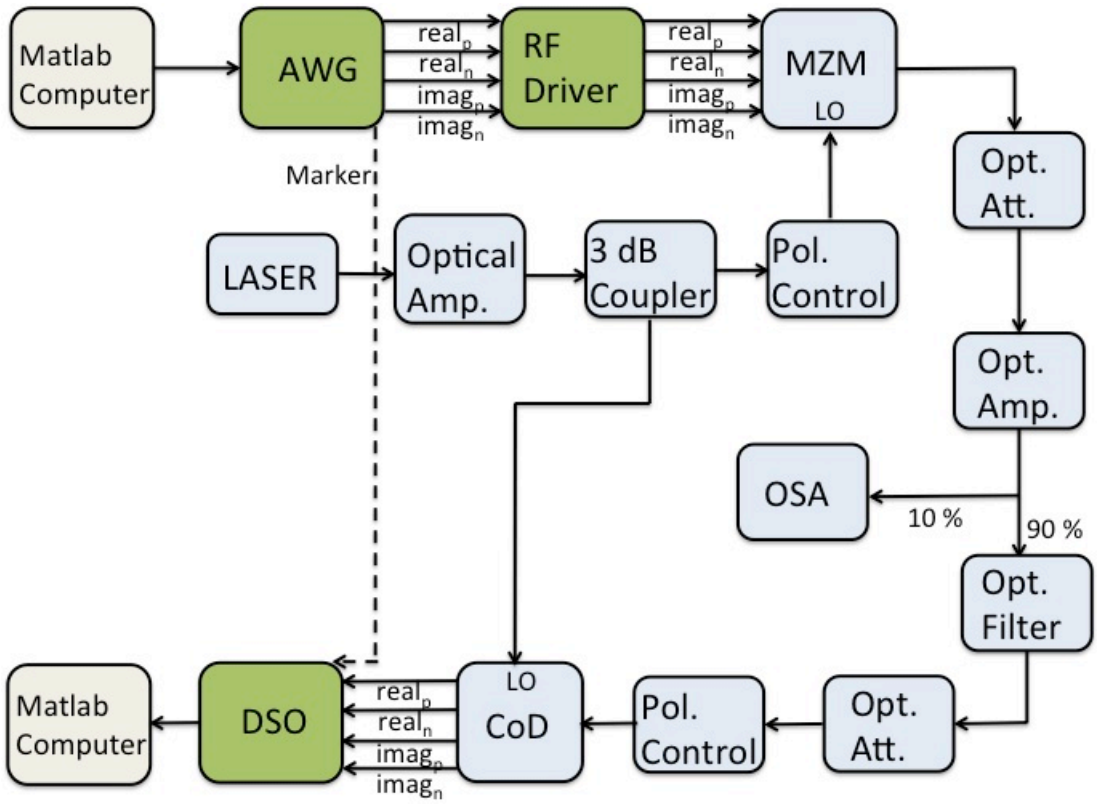

FIGURE 5.9: Configuration of Homodyne Coherent Detection. DSP processing is done offline in Matlab. Green blocks indicate analogue blocks. Light Blue blocks indicate Optical components.

- Electro-Optic Transmitter - The carrier frequency for the transmitter is generated using the external-cavity LASER (ECL) at a wavelength of 1540 nm. It is passed through optical amplifier using a polarization maintaining fiber (PMF) which amplifies it to 15 dBm power level. The 3 dB coupler divides the power input and one output is given to polarization controller. The other output is connected to local oscillator (LO) input of coherent detector. The polarization controller is used to maximize optical power at the output of optical modulator (Mach-Zender Modulator (MZM)) because the optical modulator is sensitive to the signal polarization. It is used to minimize the optical loss and maximize modulation depth in MZM. The output of polarization controller, whose power level is at 11 dBm, serves as the carrier frequency input of MZM block. The transmitted signal generated in Matlab is passed through AWG and RF Driver. The output of RF Driver is connected to modulation input of MZ Modulator, power level being around 10 dBm. The modulated optical signal is connected to optical attenuator.

- Opto-Electronic Receiver - The optical attenuator is used to vary the optical signal-to-noise ratio (OSNR). The attenuated signal is then amplified for transmission and measurement. 10% of the amplified signal is given to optical signal analyzer (OSA) for measuring OSNR of the modulated signal and 90% of the signal is used for transmission. The optical band pass filter (BPF) selects the bandwidth around the carrier frequency and the filtered output is passed through optical attenuator. The bandwidth range of optical BPF is 3 nm. The optical attenuator is used to control the maximum optical power level at the input of coherent detector. The maximum value of signal input of coherent detector is fixed at -5 dBm. The attenuated output is connected to polarization controller which allows to maximize the output level of CoD corresponding to X-polarization or Y-polarization. The LO power level is around 11 dBm and signal after coherent detection is given to digital storage oscilloscope (DSO). The sampled digital data are then transferred to computer running Matlab for offline processing.

The bias of MZM needs to be adjusted for operating it in linear region. The setting of bias is done using the first training symbol (TS_1), which has the property that it is PSK signal both in time domain and frequency domain. For adjusting the bias of MZM, TS_1 is selected and zoomed in at the DSO. The bias of MZ Modulator is now adjusted to have circular constellation for TS_1. Faithful reproduction of circular constellation at the DSO indicates that MZ Modulator is operating in the linear region and this is done on everyday before the start of the experiment, since bias drifts with temperature. During the experiment, using the optical attenuator, the OSNR value is varied from 2 to 13 dB and corresponding BER obtained by Matlab program is tabulated. Figure 5.10 shows the BER obtained for different values of OSNR for the optical B2B experiment.

Figure 5.11 shows the homodyne configuration with the addition of standard single mode optical fiber (SSMF) of length 50 km. Again OSNR value is varied and BER values

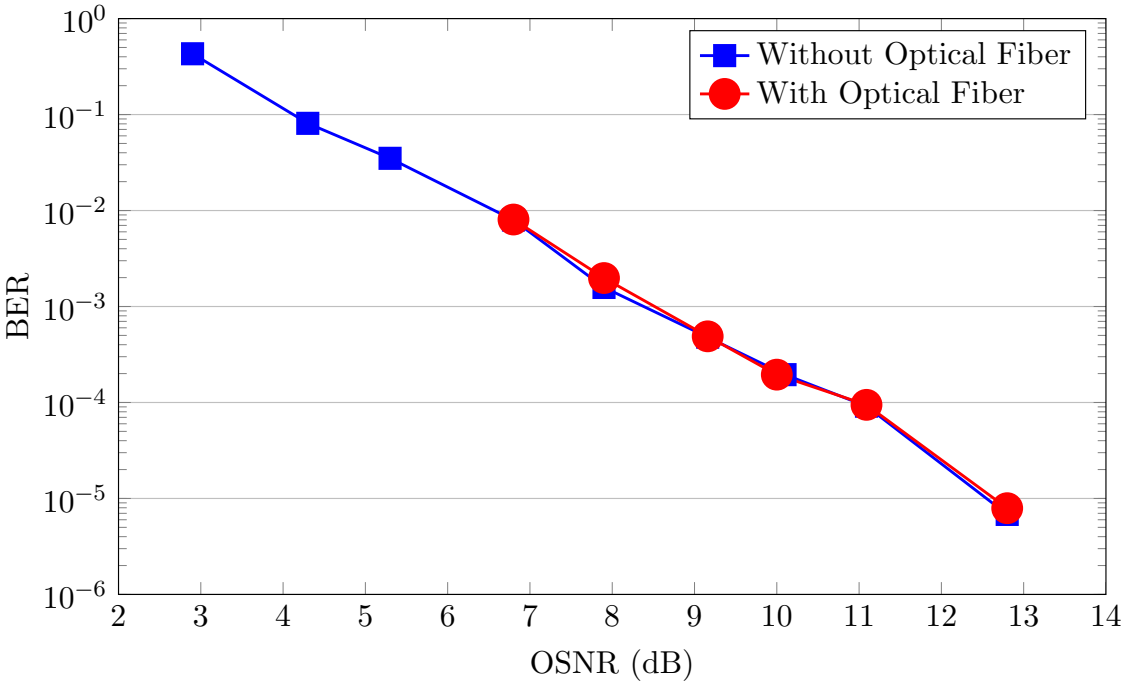

FIGURE 5.10: BER vs SNR for single-band Optical Back-to-Back Experiment

are noted. Figure 5.10 shows the BER as a function of OSNR with introduction of SSMF. The introduction of SSMF does not change the performance of the system due to presence of cyclic prefix (CP) which helps in tolerance of chromatic dispersion and is same as optical B2B configuration.

5.6 Heterodyne Coherent Detection Configuration

Figure 5.12 shows the heterodyne coherent detection configuration with separate LASER sources used for LO input of MZ Modulator and Coherent Detector. This depicts a real-world scenario where transmitter and receiver have different LASER sources. Again OSNR is varied and BER is calculated using offline Matlab receiver. Due to large variations of frequency of receiver LASER compared to transmitter LASER, it was found that frame size of 80 OFDM symbols was very long and BER obtained was very high. To overcome this problem, frame size was decreased by reducing the number of data symbols in each frame. This results in more frequent estimation of CFO using training symbols. Frame size was reduced in steps of 5 from 80 onwards. BER vs. OSNR curve was calculated at every frame size value. It was found with 45 OFDM symbol frame size, the BER vs. OSNR performance was lower than for higher frame sizes and also it remained the same when frame size was further lowered. Results of BER vs. OSNR are plotted in Figure 5.13, which validates the experimental configuration and also the full Matlab transceiver. From these optical physical layer experiments, the algorithms used were validated in presence of OSNR

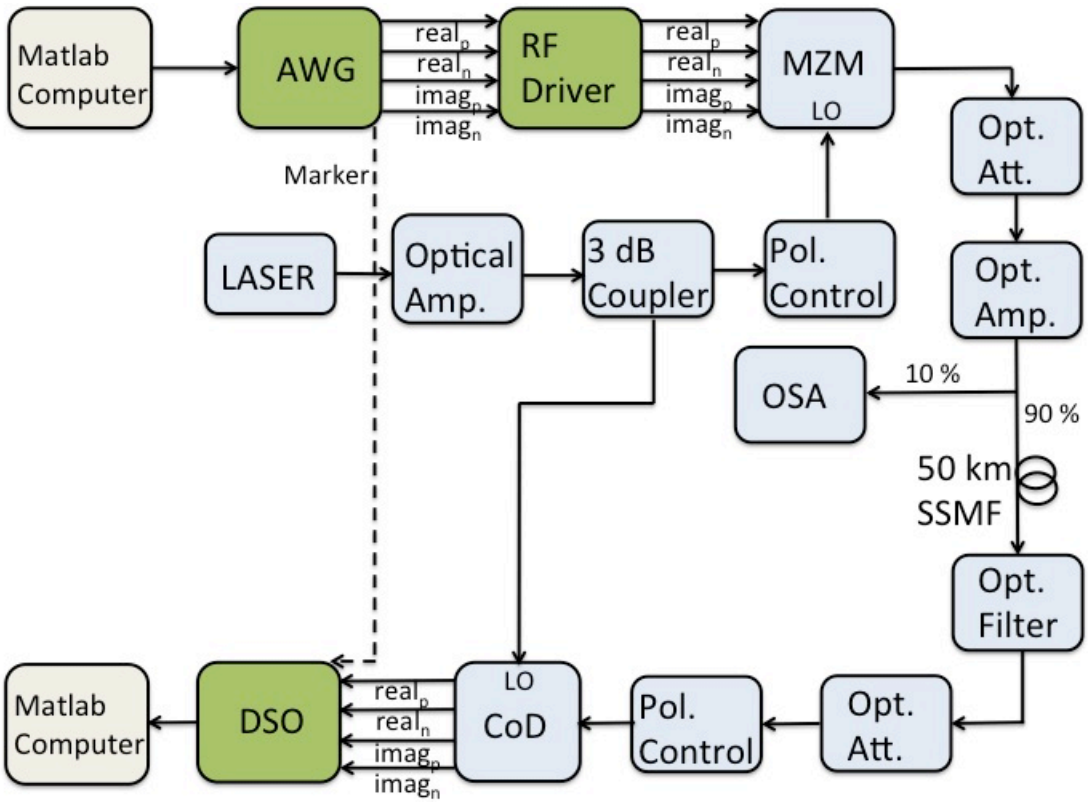

FIGURE 5.11: Configuration of Homodyne Coherent Detection with SMF of 50 km. DSP processing is done offline in Matlab. Green blocks indicate analogue blocks. Light Blue blocks indicate Optical components.

degradation and the size of frame which can be used in case of heterodyne configuration was found. This value of frame size of around 45 OFDM symbols is used in the real-time FPGA prototype platform. The DSP algorithms are realized in hardware on the real-time FPGA platform and characterization of performance of these algorithms is done in the next Section.

5.7 Real-Time FPGA Platform

A real-time FPGA platform was built as part of 100GFLEX project. Ekinops, industrial partner of the project gave us Altera FPGA platform on which to develop the DSP algorithms. It is to contain DSP blocks of transmitter and receiver in a single platform. The second platform of Xilinx was used to only interface DAC and ADC blocks and link between Altera and Xilinx FPGA was done using SFP+ interface. Hence, both Xilinx and Altera FPGA were used for the real-time FPGA prototype. The different building blocks used for real-time transmitter and receiver are explained below.

Transmitter Platform - Figure 5.14 shows the different blocks and the connections starting from bit generation to analog output. The OFDM frame format adopted in 100GFLEX project is shown in Figure 5.15. The training symbol TS_1 is used for timing

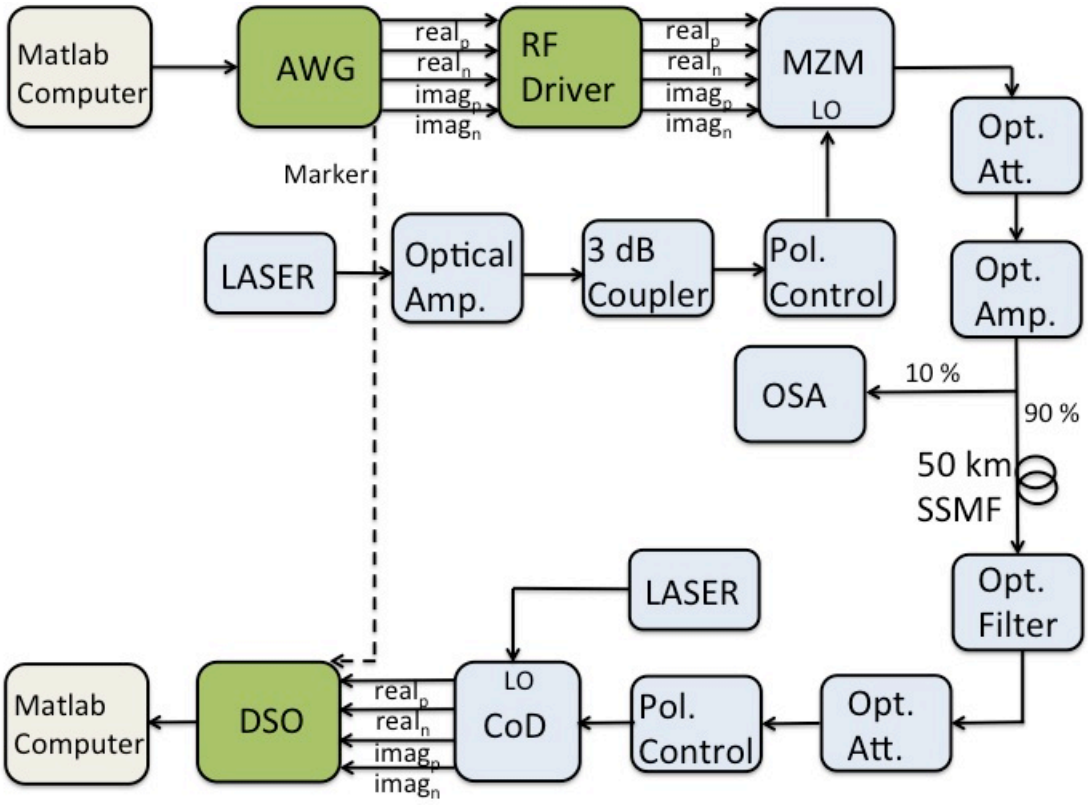

FIGURE 5.12: Configuration of Heterodyne Coherent Detection with standard single mode fiber (SSMF) of 50 km. DSP processing is done offline in Matlab. Green blocks indicate analogue blocks. Light Blue blocks indicate Optical components.

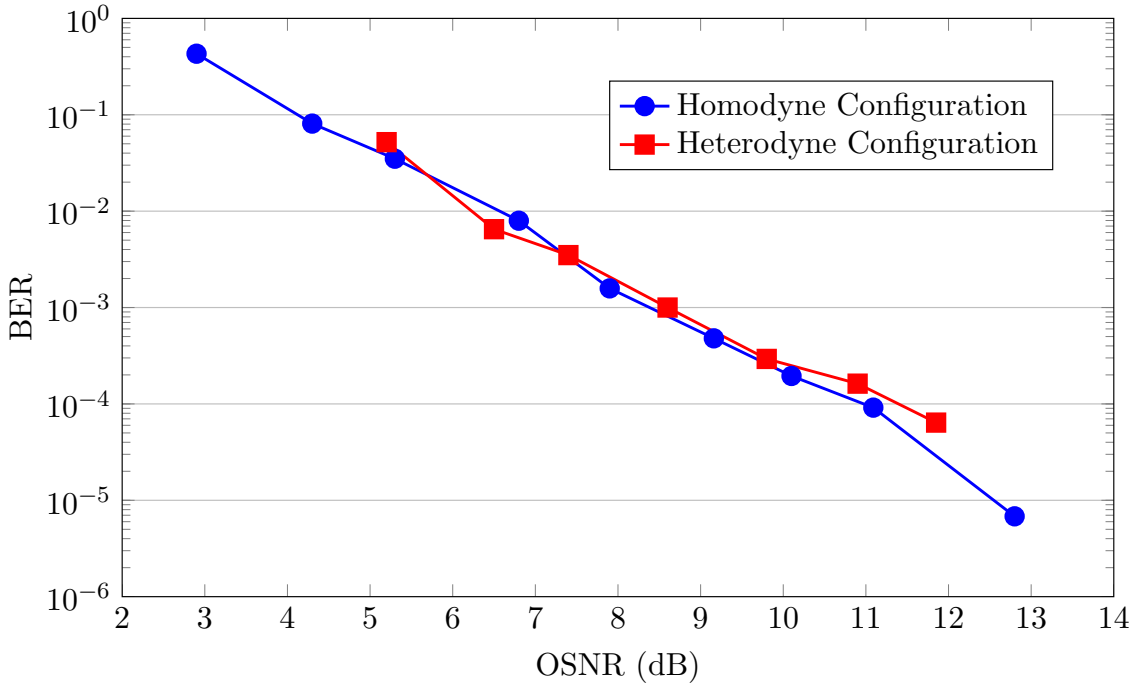

FIGURE 5.13: BER vs SNR for single-band CO-OFDM system for Heterodyne Detection

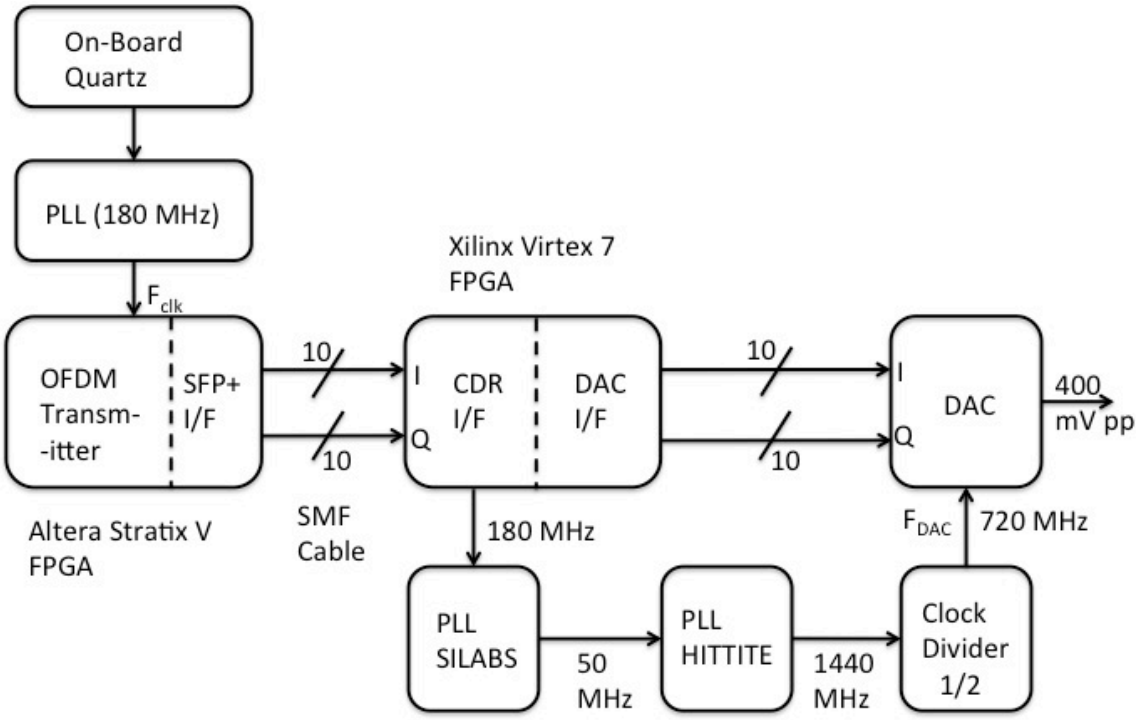

FIGURE 5.14: Real-Time FPGA Transmitter Block Diagram. PLL - Phase Locked Loop, SFP+ - Enhanced Small Form-factor Pluggable, SMF Cable - Single Mode Fiber Cable, I/F - Interface, CDR I/F - Clock Data Recovery Interface, DAC I/F - Digital-to-Analog Converter Interface.

synchronization and fractional CFO estimation. Training symbol TS_2 is used for integer CFO estimation, least squares channel estimation. Channel and Phase tracking is done using least mean squares (LMS) algorithm and common phase error (CPE) algorithm respectively. Frame contains 47 data symbols and thus totally 49 symbols in each frame. Table 5.2 shows the parameters used for TS_1, TS_2 and $DS_{1,2,..,47}$.

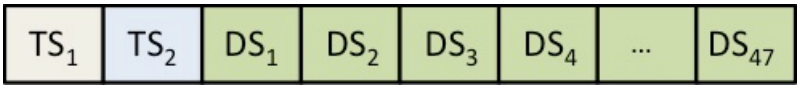

FIGURE 5.15: 100GFLEX Frame Format

The digital OFDM transceiver is implemented in an Altera Stratix V FPGA development board, which consists of Mapper, IFFT, Cyclic Prefix Addition blocks. The Stratix V FPGA used is 5SGXEA7K2F40C2. The clock for Altera FPGA transmitter is generated by on-board quartz and PLL. It operates at a clock frequency of 180 MHz. Data is generated by four Pseudo Random Binary Sequence (PRBS) generators, which are fed into four IFFTs operating at 180 MHz. The IFFT uses radix-2^2 algorithm and uses single delay feedback (SDF) architecture, providing one sample per cycle output. After addition of cyclic prefix, the four outputs are fed into SFP+(enhanced small form-factor pluggable)

Table 5.2: 100GFLEX Frame Format Parameters

Parameter	Value
Sampling Frequency (F_{DAC}, F_{ADC})	720 MHz
FPGA Clock Frequency (F_{clk})	180 MHz
Parallel Factor Used (R)	4
Cylic Prefix size (N_{cyp})	8
IFFT/FFT size (N)	256
Symbol size (N_{sym})	264
Training Symbols in each frame	2
Data Symbols in each frame	47
OFDM Symbols in each frame	49
Mapping scheme used (M)	QPSK
Sub-carrier spacing (F_{DAC}/N)	2.8125 MHz
OFDM symbol duration (T_{sym})	366.6 ns
Training Symbol TS_1	Minn-Bhargava Type
TS_1 Pattern used	[1 1 $-$1 1]
Number of Pilots per OFDM Symbol (N_p)	8
Position of Pilots	[31 63 95 127 159 191 223 255]
DAC Voltage range	1 $Vp-p$
ADC Voltage range	0.5 $Vp-p$

interface which connects Altera FPGA board and Xilinx FPGA board using single mode fiber (SMF) cable.

The Xilinx Virtex 7 development board (VC707) uses clock data recovery circuit to recover clock from the received data. The clock extracted is 180 MHz clock, which is fed into PLL (SILABS SI5338) to generate the 50 MHz reference clock for PLL (HITTITE HMC833LP6GE). The PLL (HITTITE HMC833LP6GE) generates twice the sampling frequency of 1440 MHz and gives it to a clock divider (HMC394LP4 programmable divider) circuit. Xilinx FPGA board provides four samples per cycle at 180 MHz to DAC sampling at 720 MHz. The output range of DAC (FMC204) is 1 $Vp-p$ and resolution is 10-bits. The output range used by the platform is 0.4 $Vp-p$. The real and imaginary outputs are then fed into two 3 dB attenuators to reduce the peak-to-peak voltage. This reduction is done to make the voltage to fit in the range of ADC (FMC126), which has an input range of 0.25 $Vp-p$. The output of attenuator is fed into ADC, which is equivalent to a electrical back-to-back (B2B) experiment.

Receiver Platform - Figure 5.16 shows the receiver platform both in case of synchronized sampling and asynchronized sampling by the use of switch. In case of synchronized sampling, the sampling frequency for the ADC will be given by the PLL (HITTITE HMC833LP6GE) of the transmitter chain. It will give a 1440 MHz clock to ADC clock, which has an internal divide by 2 circuit. This mode is indicated by switch value of "0". In asynchronous sampling mode (switch value of "1"), the sampling clock for ADC is generated by another PLL (HITTITE HMC833LP6GE).

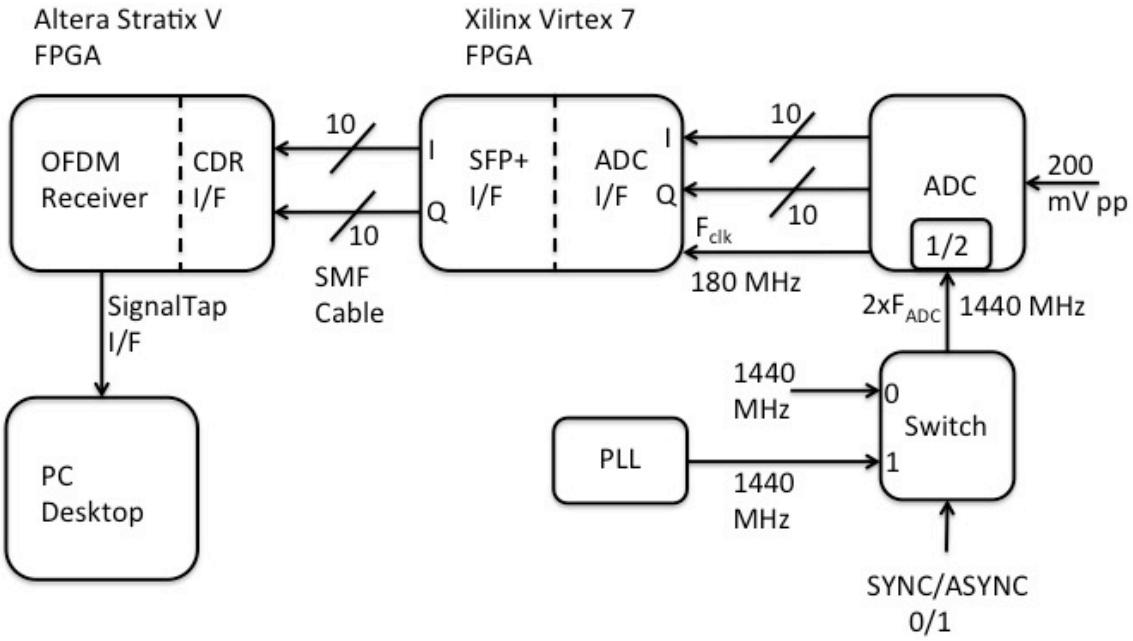

FIGURE 5.16: Real-Time FPGA Receiver Block Diagram. ADC - Analog-to-Digital Converter, I/F - Interface, SFP+ I/F - Enhanced Small Form-factor Pluggable Interface, SMF Cable - Single Mode Fiber Cable, CDR I/F - Clock Data Recovery Interface, PLL - Phase Locked Loop.

The incoming real and imaginary data are sampled at 720 MHz and given to Xilinx FPGA Development Board (VC707). The clock for Xilinx FPGA is given by the clock output of ADC, which is 180 MHz. The ADC has a resolution of 10 bits. It gives four parallel samples every cycle to Xilinx FPGA due to FPGA clock frequency being one-fourth that of ADC sampling frequency. The Xilinx FPGA then transfers four parallel real and imaginary samples to Altera FPGA Board by using SFP+ interface on SMF cable. On the Altera FPGA board side, the clock data recovery (CDR) circuit recovers the clock of 180 MHz and it is used as the clock for OFDM Receiver. The OFDM receiver consists of the blocks in the following order: Time synchronization block, fractional CFO estimation block, CFO compensation block, FFT block, integer CFO estimation block, Channel estimation and compensation block, CPE estimation block and demapper block. Altera SignalTap block is connected to sample the outputs at various points of the receiver to verify the correctness of the operation of the block.

Time Synchronization block uses the 4-Parallel block-parallel full-streaming architecture proposed in this thesis. It is slightly modified to take care of the different sign pattern used in 100GFLEX TS_1. After detection of starting point of the frame, it gives 4-parallel samples to CFO compensation block along with estimate of fractional CFO estimate. The output of CFO compensation block is given to FFT. Here, four separate FFTs are present which work in a round robin fashion to process the input samples. The FFT uses radix-2^2 algorithm and single delay feedback (SDF) architecture and produces single output per

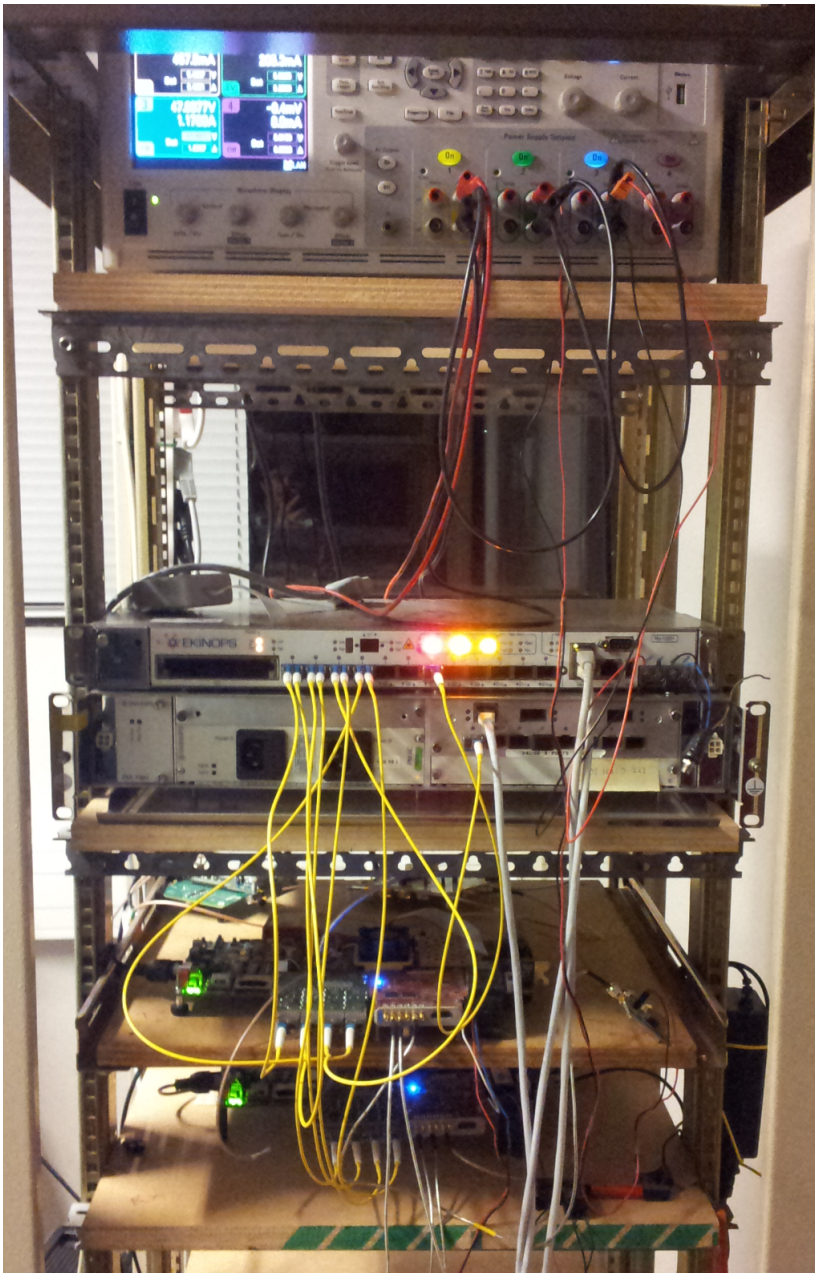

FIGURE 5.17: Snapshot of Real-Time FPGA Transceiver Platform. The topmost rack shows the power supply for the configuration, the second rack is the EKINOPS Altera FPGA Digital Transceiver, the third rack shows the Xilinx Virtex-7 FPGA interfaced to DAC board, the bottom most rack shows the other Xilinx Virtex-7 FPGA interfaced to ADC. The yellow cables are single mode fiber (SMF) cables to connect using SFP+ interface.

cycle. The output of FFT is then given to integer CFO estimation block, which uses cross-correlation with known TS_2 portion to estimate integer CFO. Then TS_2 symbol is passed through Least Squares equalizer to produce an initial channel estimate. During data symbol compensation, it uses LMS equalizer to update the channel coefficients. Finally, CPE is estimated using the pilots embedded in the OFDM symbol and compensated. Here, the architecture is not end-to-end parallel like the one proposed in Chapter 4. A snapshot of fully connected Real-time Transceiver platform is shown in Figure 5.17.

5.8 Performance of the Proposed Timing Synchronization Algorithm on Real-Time FPGA Platform

FIGURE 5.18: Altera SignalTap Snapshot of coarse synchronization output. Presence of periodic zeros indicate cyclic prefix removal and bigger gap zeros indicate the removal of the first training symbol in the output fed into FFT block.

In the Electrical B2B experiment, with no presence of CFO, the only unknown to estimate to demodulate OFDM signal is start of frame. The objective is to validate the time synchronization algorithm. Proposed block-parallel architecture for Minn-Bhargava algorithm was used for estimation of the starting point. The Verilog HDL generated using CatapultC was integrated into the setup and performance of the system was examined. To make the performance easier to observe, all the data symbols were coded the same and hence validation of the correct starting point could be done easily by observing SignalTap outputs after synchronization block. Figure 5.18 shows the output captured after timing synchronization block. The bigger zero pulses and corresponding zeros indicate TS_1 which is not passed through the FFT and small zero pulses indicate cyclic prefix removed. The gap between two large zeros indicate the time between two full frames. It can be observed that width of zero pulses remains constant which is indication of correct synchronization estimation. Figure 5.19 shows the zoomed-in version, where values of four parallel inputs

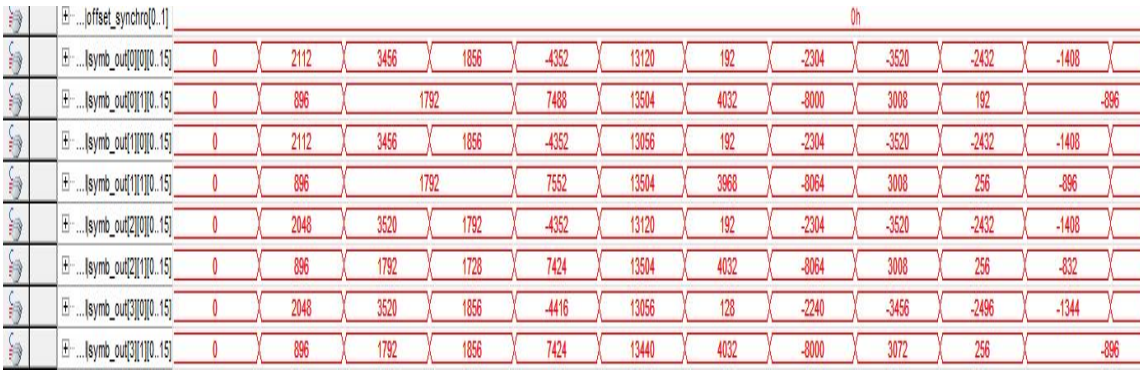

FIGURE 5.19: Altera SignalTap Snapshot of coarse synchronization output of OFDM symbols.Starting from second row, it contains real and imaginary signals alternatively. Correctness of the synchronization is verified by observing that alternate rows have repeating values indicating correct synchronization is achieved.

to FFT are indicated. The values in the four lanes has to be identical, since all data symbols have the same values. It can be again observed that the four data symbols have the same value and it is the correct starting point of the symbol. Thus, proposed synchronization algorithm is validated using real-time FPGA platform experiment. The data captured through SignalTap is then analyzed to see whether synchronization performance remains the same over large number of frames. Similar results were observed even with asynchronous sampling setup. Hence, the real-time platform setup and synchronization algorithm were validated by observing the outputs after synchronization step. BER of the system was calculated using the captured data. The data was captured repeatedly and on every acquisition five OFDM frames were captured. BER was averaged over multiple acquisition and found to be near zero.

5.9 Future Experiments proposed for Real-Time Platform

Real-time FPGA platform took a long time in setting up and a working prototype was available very recently. Hence, experiments including optical setup and FPGA platform could not be done. The two experiments remaining with the addition of optical components are:

- Homodyne Optical B2B experiment - The AWG and DSO in the offline configuration (Figure 5.9) will be replaced by FPGA transmitter (Figure 5.14) and FPGA receiver (Figure 5.16) respectively. BER is calculated using data acquired from SignalTap at different values of OSNR.

- Heterodyne Optical B2B experiment - Again AWG and DSO in the offline Heterodyne configuration (Figure 5.12 will be replaced by FPGA transmitter and receiver. BER calculation is done at various values of OSNR.

5.10 Conclusions

In this chapter, optical experiments were done to validate the frame structure and algorithms adopted for demodulating CO-OFDM signal. The experiments with real optical equipments prove the validity of the frame structure and the algorithms used. Value of frame size obtained in heterodyne configuration was used in real-time FPGA prototype platform to validate the hardware implementation of timing synchronization algorithm. Both synchronous sampling and asynchronous sampling yielded similar results. It also showed the architecture's suitability for real-time processing of optical OFDM signals. Further experiments to be performed with real-time FPGA platform is detailed and also experiments using dual-polarization CO-OFDM system to estimate and compensate for polarization effect.

Chapter 6

Conclusions and Perspectives

6.1 Overview

In this thesis, low-complexity algorithms and parallel architectures were explored for efficient realization of the digital signal processing (DSP) blocks of a CO-OFDM transceiver. To achieve the total data rate of 100 Gb/s using present day data converters (DAC and ADC) bandwidth, multi-band CO-OFDM (MB-CO-OFDM) is adopted. MB-CO-OFDM divides total bandwidth of 50 GHz into smaller sub-bands and thus bandwidth requirement of DAC/ADC is reduced significantly. Hence, the total MB-CO-OFDM architecture consists of identical transceiver chains which transmit/decode the data in both polarizations of every sub-band. The major idea is that, since identical DSP architectures are used in each polarization of every sub-band, gains obtained due to resource optimization will be multi-fold. Hence, exploration of low-complexity algorithms and parallel architectures was done for single-polarization, single-band CO-OFDM transceiver. The only block which changes from single-polarization and dual-polarization is the channel estimation block in the receiver, with rest of the DSP blocks replicated. Also, realization of architecture on FPGA platforms makes it necessary to have parallel architecture, since FPGA can reach a maximum of few hundreds of MHz, while DAC/ADC interfaced to it will be in range of GHz. Hence, scalable parallel architectures are required for every DSP block in the CO-OFDM signal processing chain to avoid costly replication to match the input sampling rate. With these set of requirements, the major contributions of this thesis are listed below.

- A novel low-complexity hierarchical time synchronization algorithm is proposed for intersymbol interference (ISI) channel. The mean square error (MSE) performance is comparable to high-efficiency cross-correlation only algorithms and computational complexity comparable to low-complexity auto-correlation algorithms. It also provides fractional carrier frequency offset (CFO) estimate.

- The low-complexity hierarchical time synchronization is modified for an optical channel. The modification reduces the computational complexity still further to a two-step process from a three-step process and attains MSE performance superior to auto-correlation algorithms.

- A scalable block-parallel architecture is proposed for efficient parallelization of auto-correlation algorithm. The required amount of resources (complex multipliers) scales linearly with number of parallel outputs. The proposed approach of parallelization is very general and can be used to parallelize any auto-correlation algorithm. Comparison with previous parallel architecture proposals show a 17 to 72% decrease in resource usage for Minn-Bhargava algorithm parallelization.

- The proposed block-parallel architecture is then modified to support the proposed hierarchical synchronization algorithm and parallelization obtained for auto-correlation and cross-correlation algorithms is reported.

- A new multipath delay (MPD) architecture for radix-2^2 IFFT/FFT algorithm is proposed which uses lesser resources compared to radix-2 architecture and is easily scalable to higher number of parallel outputs.

- An end-to-end parallel architecture for CO-OFDM system is proposed, with each block parallelized to handle multiple samples per cycle inputs. Scalability and reduction in algorithmic/architectural complexity due to use of proposed time synchronization algorithm, MPD architecture of radix-2^2 for IFFT/FFT, data representation optimization leading to savings of multipliers in case of channel estimation and common phase estimation (CPE) is shown. Fixed-point analysis of the CO-OFDM transceiver for adoption of reduced bitwidth for meeting a particular value of root mean square error (RMSE) in case of transmitter and getting BER curve as close as possible to floating-point computation in case of receiver. Since multi-band CO-OFDM (MB-CO-OFDM) for reaching total data rate of 100 Gbs/ uses the same parallel architecture for all the sub-bands, the optimization of resources in a single sub-band leads to multiplied savings in resources across all the sub-bands.

- The algorithms and the frame structure adopted are validated by experiments performed in the optical laboratory. Offline experiments using Matlab transmitter/receiver are performed to validate the algorithm performance in homodyne and heterodyne coherent detection configurations. From the heterodyne configuration, frame size suitable for use with our optical setup was found. This value of frame size was used in the development of the real-time FPGA platform. The validation of timing synchronization algorithm was done using real-time experiment in a electrical back-to-back (B2B) configuration.

6.2 Future Work

Following sub-sections indicate future work possible in experimental and algorithm domains.

6.2.1 Real-time FPGA platform experiments

As mentioned in Chapter 5, optical experiments need to be conducted using the real-time FPGA platform. Firstly, homodyne configuration is performed to check the performance of the digital receiver only in presence of timing and phase offset. BER vs. OSNR curve will characterize the system performance. Next, in heterodyne configuration the effect of carrier frequency offset (CFO) will be seen on the BER curve. This configuration will exercise all the DSP blocks in the receiver chain. Finally, a very long length of standard single mode fiber (SSMF) (1000 km) will be connected to reach the complete real-world scenario for single-polarization single-band CO-OFDM system.

6.2.2 Dual-polarization CO-OFDM System

All the experiments in this Thesis have been conducted using only one polarization. Since coherent detection allows both polarizations to be used and thus higher utilization of the optical channel, transition to dual polarization single-band CO-OFDM system is necessary from throughput perspective. From a signal processing perspective, all the blocks except channel estimation will be replicated for the polarization added and frame structure needs to be changed slightly to accommodate for training symbols for channel estimation for both polarizations. Offline experiments conducted first to validate the channel estimation algorithm and then with real-time platform and long length of SSMF to reach the complete real-world scenario.

6.2.3 Time Domain Sampling Clock Offset (SCO) Algorithm

A low-complexity sampling clock offset (SCO) tracking algorithm and architecture in time domain will be very beneficial to control the sampling clock of the ADC and reduce the errors due to SCO offset. Position of SCO estimation in time domain reduces the loop delay for estimation and compensation compared to presently available SCO estimation algorithms [56] in frequency domain which is after FFT block. Due to presence of SCO estimation block after FFT, the delay of the feedback path is long and cannot adapt to sudden changes.

6.3 Scaling to more than 100 Gb/s with MB-CO-OFDM system

Scaling to data rate higher than 100 Gb/s like 400 Gb/s or 1 Tb/s can be achieved by increasing the size of FFT/IFFT to achieve higher data rate per polarization of a

single-band. Presently, the maximum size of FFT/IFFT is limited by the LASER phase noise variation to values less than or equal to 256. Digital CPE estimation algorithms assume that phase noise is constant across a single OFDM symbol, which breaks in case of larger OFDM symbol. RF-Pilot phase noise estimation scheme [57] has been proposed to overcome this. But the method is computationally very complex. A phase noise estimation scheme [58] which can use both RF-based pilot scheme and CPE method to handle large OFDM symbol size and still be computationally efficient would enable very high rates. The next option is when higher resolution DAC/ADC signal converters become available, it will be possible to support higher constellations in sub-carriers namely 16-QAM and 64-QAM.

Publications

Journal Publications

1. P. Udupa, O. Sentieys and L. Bramerie,"A Scalable Parallel Architecture for Coarse Time Synchronization for Coherent Optical-OFDM Systems," submitted to *IEEE Transaction Briefs on Very Large Scale Integration (VLSI) Systems*, 2014.

International Conference Publications

1. P. Udupa, O. Sentieys and P. Scalart,"A Novel Hierarchical Low Complexity Synchronization Method for OFDM Systems," in *IEEE VTC-Spring 2013*, 1-5,June 2013.

2. P. Udupa, O. Sentieys and P. Scalart,"A Block-Parallel Architecture for Initial and Fine Synchronization in OFDM Systems," in *IEEE ICC 2013*, 4761-4765, June 2013.

National Conference/Colloquium Publications

1. P. Udupa, O. Sentieys and L. Bramerie,"Design and Implementation of DSP algorithms for 100 Gbps Optical OFDM System," in *GRETSI 2013*, September 2013.

2. P. Udupa, O. Sentieys and L. Bramerie,"Design and Real Time FPGA Prototyping of 100Gb/s Optical MB-OFDM System and Beyond," in *GDR SOC/SIP 2012*, June 2012.

Bibliography

[1] Cisco VNI Forecasting Widget. [Online]. Available: http://ciscovni.com

[2] R. Essiambre, G. Kramer, P. Winzer, G. Foschini, and B. Goebel, "Capacity Limits of Optical Fiber Networks," *Journal of Lightwave Technology*, vol. 28, no. 4, pp. 662–701, 2010.

[3] K. Roberts, D. Beckett, D. Boertjes, J. Berthold, and C. Laperle, "100G and Beyond with Digital Coherent Signal Processing," *IEEE Communications Magazine*, vol. 48, no. 7, pp. 62–69, July 2010.

[4] M. Puschel, J. Moura, J. Johnson, D. Padua, M. Veloso, B. Singer, J. Xiong, F. Franchetti, A. Gacic, Y. Voronenko, K. Chen, R. Johnson, and N. Rizzolo, "SPIRAL: Code Generation for DSP Transforms," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 232–275, 2005.

[5] Garrido, M. and Grajal, J. and Sanchez, M.A. and Gustafsson, O., "Pipelined Radix-2^k Feedforward FFT Architectures," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 1, pp. 23–32, January 2013.

[6] J. D'Ambrosia, "100 Gigabit Ethernet and Beyond," *IEEE Communications Magazine*, vol. 48, no. 3, pp. S6–S13, March 2010.

[7] E. Ip, A. Lau, D. Barros, and J. Kahn, "Coherent Detection in Optical Fiber Systems," *Optics Express*, vol. 16, no. 2, pp. 753–791, 2008.

[8] S. Jansen, "Optical OFDM, a hype or is it for real?" in *Optical Communication, 2008. ECOC 2008. 34th European Conference on*, sept. 2008, p. 1.

[9] M. Taylor, "Coherent Detection for Fiber Optic Communications using Real Time Digital Signal Processing," in *Optical Fiber Communication and the National Fiber Optic Engineers Conference, 2007. OFC/NFOEC 2007. Conference on*, 2007, pp. 1–3.

[10] Y. Liu and P. Fan, "Modified Chu sequences with smaller alphabet size," *Electronics Letters*, vol. 40, no. 10, pp. 598–599, May 2004.

[11] P. Udupa, O. Sentieys and P.Scalart, "A Novel Hierarchical Low Complexity Synchronization Method for OFDM Systems," in *IEEE 77th Vehicular Technology Conference*, June 2013, pp. 1–5.

[12] S. Beyme and C. Leung, "Efficient computation of DFT of Zadoff-Chu sequences," *Electronics Letters*, vol. 45, no. 9, pp. 461–463, April 2009.

[13] S. P. Lloyd, "Least Squares Quantization in PCM," in *IEEE Transactions on Information Theory*, vol. 2, no. 28, 1982, pp. 129–137.

[14] N. Kaneda, Q. Yang, X. Liu, S. Chandrasekhar, W. Shieh, and Y.-K. Chen, "Real-Time 2.5 GS/s Coherent Optical Receiver for 53.3-Gb/s Sub-Banded OFDM," *Journal of Lightwave Technology*, vol. 28, no. 4, pp. 494–501, February 2010.

[15] X. Yi, W. Shieh, and Y. Tang, "Phase estimation for coherent optical ofdm," *Photonics Technology Letters, IEEE*, vol. 19, no. 12, pp. 919 –921, june15, 2007.

[16] S. Jansen, B. Spinnler, I. Morita, S. Randel, and H. Tanaka, "100GbE: QPSK versus OFDM," *Optical Fiber Technology*, vol. 15, no. 5-6, pp. 407–413, 2009.

[17] J. Armstrong, "OFDM for Optical Communications," *Journal of Lightwave Technology*, vol. 27, no. 3, pp. 189–204, February 2009.

[18] W. Shieh, X. Yi, Y. Ma, and Q. Yang, "Coherent Optical OFDM: has its time come?[Invited]," *Journal of Optical Networking*, vol. 7, no. 3, pp. 234–255, 2008.

[19] J. Bingham, "Multicarrier Modulation for Data Transmission: An Idea Whose Time Has Come," *IEEE Communications Magazine*, vol. 28, no. 5, pp. 5–14, May 1990.

[20] Melle, S. and Jaeger, J. and Perkins, D. and Vusirikala, V., "Market Drivers and Implementation Options for 100-GBE Transport over the WAN," *IEEE Communications Magazine*, vol. 45, no. 11, pp. 18–24, 2007.

[21] Calypto Design Systems Inc. (2011). [Online]. Available: http://calypto.com/en/products/catapult/catapult_sl

[22] W. Shieh, H. Bao, and Y. Tang, "Coherent Optical OFDM: Theory and Design," *Optics Express*, vol. 16, no. 2, pp. 841–859, 2008.

[23] B.-J. Choi, E.-L. Kuan, and L. Hanzo, "Crest-Factor Study of MC-CDMA and OFDM," in *IEEE VTS 50th Vehicular Technology Conference*, vol. 1, September 1999 - Fall, pp. 233–237.

[24] P. Liu and Y. Bar-Ness, "Closed-Form Expressions for BER Performance in OFDM Systems with Phase Noise," in *IEEE International Conference on Communications*, vol. 12, june 2006, pp. 5366–5370.

[25] L. Tomba, "On the Effect of Wiener Phase Noise in OFDM Systems," *IEEE Transactions on Communications*, vol. 46, no. 5, pp. 580–583, May 1998.

[26] Shousheng He and Torkelson, M., "Design and Implementation of a 1024-point Pipeline FFT Processor," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, 1998, pp. 131–134.

[27] Sanchez, M.A. and Garrido, M. and Lopez-Vallejo, M. and Grajal, J., "Implementing FFT-based Digital Channelized Receivers on FPGA Platforms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 4, pp. 1567–1585, 2008.

[28] Johnston, J. A., "Parallel Pipeline Fast Fourier Transformer," *IEE Proceedings for Communications, Radar and Signal Processing*, vol. 130, no. 6, pp. 564–572, 1983.

[29] R. Schmogrow, M. Winter, D. Hillerkuss, B. Nebendahl, S. Ben-Ezra, J. Meyer, M. Dreschmann, M. Huebner, J. Becker, C. Koos, W. Freude, and J. Leuthold, "Real-time OFDM Transmitter beyond 100 Gbit/s," *Opt. Express*, vol. 19, no. 13, pp. 12 740–12 749, June 2011.

[30] B. Inan, O. Karakaya, P. Kainzmaier, S. Adhikari, S. Calabro, V. A. J. M. Sleiffer, N. Hanik, and S. Jansen, "Realization of a 23.9 Gb/s real time Optical-OFDM Transmitter with a 1024 point IFFT," in *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2011 and the National Fiber Optic Engineers Conference*, 2011, pp. 1–3.

[31] Inan, B. and Adhikari, S. and Karakaya, O. and Kainzmaier, P. and Mocker, M. and von Kirchbauer, H. and Hanik, N. and Jansen, S.L., "Realization of a real-time 93.8-Gb/s polarization-multiplexed OFDM transmitter with 1024-point IFFT," in *Optical Communication (ECOC), 2011 37th European Conference and Exhibition on*, 2011, pp. 1–3.

[32] Y. M. S Chen, Q Yang and W. Shieh, "Multi-Gigabit Real-Time Coherent Optical OFDM Receiver," in *Proc. OFC/NFOEC, San Diego, CA*, March 2009, pp. 1–3.

[33] T. Schmidl and D. Cox, "Robust Frequency and Timing Synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1613–1621, December 1997.

[34] H. Minn, V. Bhargava, and K. Letaief, "A Robust Timing and Frequency Synchronization for OFDM Systems," *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, pp. 822–839, July 2003.

[35] K. Shi and E. Serpedin, "Coarse Frame and Carrier Synchronization of OFDM Systems: A New Metric and Comparison," *IEEE Transactions on Wireless Communications*, vol. 3, no. 4, pp. 1271–1284, July 2004.

[36] B. Park, H. Cheon, C. Kang, and D. Hong, "A Novel Timing Estimation Method for OFDM Systems," *IEEE Communications Letters*, vol. 7, no. 5, pp. 239–241, May 2003.

[37] S. D. Choi, J. M. Choi, and J. H. Lee, "An Initial Timing Offset Estimation Method for OFDM Systems in Rayleigh Fading Channel," in *IEEE 64th Vehicular Technology Conference*, September 2006, pp. 1–5.

[38] E. Zhou, X. Hou, Z. Zhang, and H. Kayama, "A Preamble Structure and Synchronization Method Based on Central-Symmetric Sequence for OFDM," in *IEEE Vehicular Technology Conference*, 2008, pp. 1478–1482.

[39] Yun Hee Kim and Young-Kwon Hahm and Hye Jung Jung and Iickho Song, "An Efficient Frequency Offset Estimator for Timing and Frequency Synchronization in OFDM Systems," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 1999, pp. 580–583.

[40] Chiueh, Tzi-Dar and Tsai, Pei-Yun and Lai, I-Wei, *Baseband Receiver Design for Wireless MIMO-OFDM Communications*. John Wiley and Sons Singapore Pte. Ltd., 2007, ch. 7, pp. 167–208.

[41] M. Morelli, C.-C. Kuo, and M.-O. Pun, "Synchronization Techniques for Orthogonal Frequency Division Multiple Access (OFDMA): A Tutorial Review," *Proceedings of the IEEE*, vol. 95, no. 7, pp. 1394–1427, July 2007.

[42] M. Speth, F. Classen, and H. Meyr, "Frame Synchronization of OFDM Systems in Frequency Selective Fading Channels," in *IEEE 47th Vehicular Technology Conference*, vol. 3, May 1997, pp. 1807–1811.

[43] T. Pollet, M. Van Bladel, and M. Moeneclaey, "BER Sensitivity of OFDM Systems to Carrier Frequency Offset and Wiener Phase Noise," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 191–193, 1995.

[44] J. van de Beek, M. Sandell, and P. Borjesson, "ML estimation of time and frequency offset in ofdm systems," *IEEE Transactions on Signal Processing*, vol. 45, no. 7, pp. 1800–1805, july 1997.

[45] M. Morelli and U. Mengali, "An Improved Frequency Offset Estimator for OFDM Applications," in *Communication Theory Mini-Conference*, June 1999, pp. 106–109.

[46] T. Bhatt, V. Sundaramurthy, J. Zhang, and D. McCain, "Initial Synchronization for 802.16e Downlink," in *Fortieth Asilomar Conference on Signals, Systems and Computers*, November 2006, pp. 701–706.

[47] "Third-generation partnership project ts 36.211,physical channels and modulation (release 8) technical specification group," Radio Access Network:Evolved Universal Terrestrial Radio Access (E-UTRAN), Tech. Rep., 2008.

[48] P. Serena, M. Bertolini and A. Vannucci. (2009) Optilux Toolbox. [Online]. Available: http://www.optilux.sourceforge.net

[49] P. Udupa, O. Sentieys and P.Scalart, "A Block-Parallel Architecture for Initial and Fine Synchronization in OFDM Systems," in *IEEE International Conference on Communications (ICC)*, 2013, pp. 4761–4765.

[50] V. C. Kurapati, "Analysis of IP Based Implementations of Adders and Multipliers in Submicron and Deep Submicron Technologies," Master of Science, Oklahoma State University, December 2008.

[51] Sander L. Jansen and Itsuro Morita and Noriyuki Takeda and Hideaki Tanaka, "20-Gb/s OFDM Transmission over 4,160-km SSMF Enabled by RF-Pilot Tone Phase Noise Compensation," in *Optical Fiber Communication Conference and Exposition and The National Fiber Optic Engineers Conference*. Optical Society of America, 2007.

[52] W. Shieh, Q. Yang, and Y. Ma, "107 Gb/s Coherent Optical OFDM Transmission over 1000-km SSMF Fiber using Orthogonal Band Multiplexing," *Optics express*, vol. 16, no. 9, pp. 6378–6386, 2008.

[53] R. Bouziane, P. Milder, R. Koutsoyannis, Y. Benlachtar, J. Hoe, M. Glick, and R. Killey, "Dependence of Optical OFDM Transceiver ASIC Complexity on FFT size," in *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2012 and the National Fiber Optic Engineers Conference*, 2012, pp. 1–3.

[54] Shousheng He and Torkelson, M., "A New Approach to Pipeline FFT Processor," in *The 10th International Parallel Processing Symposium*, 1996, pp. 766–770.

[55] P. Udupa, O. Sentieys and L.Bramerie, "Design and Implementation of DSP algorithms for 100Gbps Optical OFDM System," in *XXIV Colloque GRETSI*, September 2013.

[56] Lee, D. and Kyungwhoon Cheun, "A New Symbol Timing Recovery Algorithm for OFDM Systems," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 3, pp. 767–775, August 1997.

[57] S. Randel, S. Adhikari, and S. Jansen, "Analysis of RF-Pilot-Based Phase Noise Compensation for Coherent Optical OFDM Systems," *Photonics Technology Letters, IEEE*, vol. 22, no. 17, pp. 1288 –1290, sept.1, 2010.

[58] S.Hussin, K.Puntsri and R.Noe, "Improvement of RF-Pilot Phase Noise Compensation for Coherent Optical OFDM Systems via CPE Equalizer," 2013.

Résumé en français : Optique Cohérente-OFDM (CO-OFDM) a été proposée comme un candidat viable pour 100 Gigabit Ethernet (100 GbE) nœud. CO-OFDM que tout traitement à l'aide de signaux numériques de traitement (DSP) algorithmes pour estimer et compenser tous les non-idéalités de canal et opto-électronique les systèmes d'extrémité avant. Dans cette thèse, les algorithmes de faible complexité, les architectures parallèles évolutives pour grands blocs de calcul complexe de CO-OFDM émetteur-récepteur sont explorées. Un temps faible complexité synchronisation est proposé qui donne de meilleures performances que algorithmes d'auto-corrélation de canal optique. Une architecture parallèle évolutive est proposé pour l'algorithme qui peut prendre en charge plusieurs échantillons parallèles et réduit l'utilisation des ressources de l'ordre de 70% par rapport à la proposition précédente. Un parallèle bout-à-bout CO-OFDM l'architecture d'émetteur-récepteur est proposé qui intègre parallèlement à radix-2^2 bloc IFFT/FFT, ce qui réduit considérablement la complexité de calcul par rapport à radix-2 architecture et canal bloc d'estimation qui utilise la représentation des données optimisations pour supprimer multiplicateurs, entraînant des gains de 24% de la région. Enfin, les algorithmes et architectures ont été validés par des expériences hors ligne/Matlab et FPGA en temps réel la plate-forme expériences, respectivement.

Mots clés : Optique cohérente-OFDM, la synchronisation du temps/fréquence, algorithmes de faible complexité, architectures parallèles, fibre optique

Résumé en anglais : Coherent Optical-OFDM (CO-OFDM) has been proposed as a viable candidate for 100 gigabit Ethernet (100GbE) node. CO-OFDM does all processing using digital signal processing (DSP) algorithms to estimate and compensate all non-idealities of channel and opto-electronic front end systems. In this thesis, low-complexity algorithms, scalable parallel architectures for major computationally complex blocks of CO-OFDM transceiver are explored. A low-complexity time synchronization is proposed which gives better performance than auto-correlation algorithms in optical channel. A scalable parallel architecture is proposed for the algorithm which can support multiple parallel samples and reduces resource usage of around 70% compared to previous proposal. An end-to-end parallel CO-OFDM transceiver architecture is proposed which incorporates parallel radix-2^2 IFFT/FFT block, which reduces the computational complexity significantly compared to radix-2 architecture and channel estimation block which uses data representation optimizations to remove multipliers, resulting in area gains of 24%. Finally, the algorithms and architectures were validated using offline/Matlab experiments and real-time FPGA platform experiments respectively.

Keywords : Coherent Optical-OFDM, time/frequency synchronization, low-complexity algorithms, parallel architectures, optical fiber