Thesis for the Degree of Doctor of Philosophy

Implementation and Evaluation of Signal Processing Circuits for Optical Communication

Erik Börjeson

Department of Computer Science and Engineering
Chalmers University of Technology
Göteborg, Sweden, 2024

Implementation and Evaluation of Signal Processing Circuits for Optical Communication
Erik Börjeson
ISBN: 978-91-8103-015-0

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr: 5473
ISSN: 0346-718X

Department of Computer Science and Engineering
Chalmers University of Technology
SE–412 96 Göteborg
Sweden
Telephone: +46 (0)31–772 10 00

Printed in Sweden by
Chalmers digitaltryck
Chalmers Tekniska Högskola
Göteborg, Sweden, 2024

Implementation and Evaluation of Signal Processing Circuits for Optical Communication

Erik Börjeson
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

The digital signal processing (DSP) circuits used in the fiber-optic communication links that make up the backbone of the Internet can be a significant contributor the over-all power dissipation of a link. As the number of connected users and their bandwidth requirements are expected to continue to grow over the coming years, the development of power-efficient high-throughput DSP systems is a critical factor in enabling this growth. Unfortunately, DSP designers can no longer depend on foundries delivering faster and more power-efficient circuits for each new process node, due to both economical and physical limitations. As a result, more stringent speed and power requirements are put on the circuit designs.

Carrier phase recovery (CPR) is one subsystem of a typical DSP system for fiber-optic communication. In this thesis, we explore and evaluate circuit designs of multiple types of CPR, with a focus on single-mode systems. The circuit designs allow us to uncover trade-offs between power dissipation, area, throughput and signal degradation, for different types of systems employing a range of modulation formats. Coupled-core multi-mode fiber systems have been suggested as a way to increase throughput by utilizing also the spatial dimension, and this thesis describes a multiple-input multiple-output adaptive equalizer targeting these systems. The equalizer circuit enables exploration of how this critical subsystem scales to higher core counts. Additionally, we describe a circuit verification and evaluation environment that has the potential to speed up simulations by orders of magnitude by emulating a fiber-optic link onboard an application-specific integrated circuit or a field-programmable gate array.

Keywords: Adaptive Equalization, Application-Specific Integrated Circuits, Carrier Phase Recovery, Communication Systems, Digital Signal Processing, Fiber-Optic Communication

Publications

This thesis is based on the work contained in the following publications:

[I] **E. Börjeson**, C. Fougstedt, and P. Larsson-Edefors, "VLSI Implementations of Carrier Phase Recovery Algorithms for M-QAM Fiber-Optic Systems", *Journal of Lightwave Technology*, vol. 38, no. 14, pp. 3616–3623, July 2020.

[II] **E. Börjeson**, and P. Larsson-Edefors, "Energy-Efficient Implementation of Carrier Phase Recovery for Higher-Order Modulation Formats", *Journal of Lightwave Technology*, vol. 39, no. 2, pp. 505–510, Jan. 2021.

[III] **E. Börjeson** and P. Larsson-Edefors, "Benchmarking of Carrier Phase Recovery Circuits for M-QAM Coherent Systems", *Optical Fiber Communication Conference (OFC)*, p. W6A.17, June 2021.

[IV] **E. Börjeson**, and P. Larsson-Edefors, "Multi-Format Carrier Phase Recovery Using a Programmable Circuit", *Signal Processing in Photonic Communications (SPPCom)*, p. SpTu4D.2, July 2021.

[V] **E. Börjeson**, E. Deriushkina, M. Karlsson, M. Mazur, and P. Larsson-Edefors, "Circuit Implementation of Pilot-Based Dynamic MIMO Equalization for Coupled-Core Fibers", *Optical Fiber Communication Conference (OFC)*, p. W1E.4, Mar. 2024.

[VI] **E. Börjeson**, and P. Larsson-Edefors, "Fiber-on-Chip: Digital Emulation of Channel Impairments for Real-Time DSP Evaluation", *Journal of Lightwave Technology*, vol. 41, no. 3, pp. 888–896, Feb. 2023.

Related work by the author not included in this thesis:

[A] L. Lundberg, **E. Börjeson**, C. Fougstedt, M. Mazur, M. Karlsson, P. Andrekson, and P. Larsson-Edefors, "Power Consumption Savings Through Joint Carrier Recovery for Spectral and Spatial Superchannels", *European Conference on Optical Communication (ECOC)*, p. We2.26, Sept. 2018,

[B] **E. Börjeson**, C. Fougstedt, and P. Larsson-Edefors, "ASIC Design Exploration of Phase Recovery Algorithms for M-QAM Fiber-Optic Systems", *Optical Fiber Communication Conference (OFC)*, p. W3H.7, Mar. 2019.

[C] **E. Börjeson**, C. Fougstedt, and P. Larsson-Edefors, "Towards FPGA Emulation of Fiber-Optic Channels for Deep-BER Evaluation of DSP Implementations", *Signal Processing in Photonic Communications (SPPCom)*, p. SpTh1E.4, July 2019. [Best Student Paper Award]

[D] P. Larsson-Edefors, and **E. Börjeson**, "Power-Efficient ASIC Implementation of DSP Algorithms for Coherent Optical Communication", *IEEE Photonics Society Summer Topical Meeting Series (SUM)*, p. MA1.1, July 2020. [Invited]

[E] **E. Börjeson**, and P. Larsson-Edefors, "Cycle-Slip Rate Analysis of Blind Phase Search DSP Circuit Implementations", *Optical Fiber Communication Conference (OFC)*, p. M4J.3, Mar. 2020.

[F] C. Fougstedt, O. Gustafsson, C. Bae, **E. Börjeson**, and P. Larsson-Edefors, "ASIC Design Exploration of DSP and FEC of 400-Gbit/s Coherent Data-Center Interconnect Receivers", *Optical Fiber Communication Conference (OFC)*, p. Th2A.38, Mar. 2020.

[G] H. Kan, H. Zhou, **E. Börjeson**, M. Karlsson, and P. Larsson-Edefors, "Digital Emulation of Time-Varying PMD for Real-Time DSP Evaluations", *Asia Communications and Photonics Conference (ACP)*, p. M4H.4, Oct. 2021,

[H] K. Liu, **E. Börjeson**, C. Häger, and P. Larsson-Edefors, "FPGA-based Optical Kerr Effect Emulator", *Signal Processing in Photonic Communications (SPPCom)*, p. SpTh1I.2, July 2022,

[I] M. Mazur, J. C. Castellanos, R. Ryf, **E. Börjeson**, T. Chodkiewicz, V. Kamalov, S. Yin, N. K. Fontaine, H. Chen, L. Dallachiesa, S. Corteselli, P. Copping, J. Gripp, A. Mortelette, B. Kowalski, R. Dellinger, D. T. Neilson, and P. Larsson-Edefors, "Transoceanic Phase and Polarization Fiber Sensing Using Real-Time Coherent Transceiver", *Optical Fiber Communication Conference (OFC)*, p. M2F.2, Mar. 2022.

[J] P. Larsson-Edefors, and **E. Börjeson**, "Fiber-on-Chip: Digital FPGA Emulation of Channel Impairments for Real-Time Evaluation of DSP", *Optical Fiber Communication Conference (OFC)*, p. W3H.3, Mar. 2022. [Invited]

[K] M. Mazur, L. Dallachiesa, N. K. Fontaine, R. Ryf, **E. Börjeson**, H. Chen, H. Sakuma, T. Ohtsuka, T. Hayashi, T. Hasegawa, H. Tazawa, D. T. Neilson, and P. Larsson-Edefors, "Real-Time Transmission over 2x55 km All 7-Core Coupled-Core Multi-Core Fiber Link" *Optical Fiber Communication Conference (OFC)*, p. Th4A.1, Mar. 2022. [Postdeadline paper]

[L] M. Mazur, R. Ryf, N. K. Fontaine, A. Marotta, **E. Börjeson**, L. Dallachiesa, H. Chen, T. Hayashi, T. Nagashima, T. Nakanishi, T. Morishima, F. Graziosi, L. Palmieri, D. T. Neilson, P. Larsson-Edefors, A. Mecozzi, and C. Antonelli, "Real-Time MIMO Transmission over Field-Deployed Coupled-Core Multi-Core Fibers" *Optical Fiber Communication Conference (OFC)*, p. Th4B.8, Mar. 2022. [Postdeadline paper]

[M] M. Mazur, N. Parkin, R. Ryf, A. Iqbal, P. Wright, K. Farrow, N. K. Fontaine, **E. Börjeson**, K. W. Kim, L. Dallachiesa, H. Chen, P. Larsson-Edefors, A. Lord, and D. Neilson, "Continuous Fiber Sensing over Field-Deployed Metro Link using Real-Time Coherent Transceiver and DAS" *European Conference on Optical Communication (ECOC)*, p. Mo4A.2, Sept. 2022,

[N] R. R. Sagredo, **E. Börjeson**, A. Mirani, M. Karlsson, and P. Larsson-Edefors, "Waveform Memory for Real-Time FPGA Test of Fiber-Optic Receiver DSPs" *IEEE Nordic Circuits and Systems Conference (NORCAS)*, Oct. 2022.

[O] K. Liu, **E. Börjeson**, C. Häger, and P. Larsson-Edefors "FPGA Implementation of Multi-Layer Machine Learning Equalizer with On-Chip Training", *Optical Fiber Communication Conference (OFC)*, p. M1F.4, Mar. 2023.

[P] M. Mazur, D. Wallberg, L. Dallachiesa, **E. Börjeson**, R. Ryf, M. Bergroth, B. Josefsson, N. K. Fontaine, H. Chen, D. T. Neilson, J. Schröder, P. Larsson-Edefors, and M. Karlsson, "Field Trial of FPGA-Based Real-Time Sensing Transceiver over 524 km of Live Aerial Fiber", *Optical Fiber Communication Conference (OFC)*, p. Tu3G.4, Mar. 2023.

[Q] **E. Börjeson**, C. Häger, K. Liu, and P. Larsson-Edefors, "Real-Time Implementation of Machine-Learning DSP", *Optical Fiber Communication Conference (OFC)*, p. Th3J.1, Mar. 2024. [Invited]

[R] M. Mazur, D. Wallberg, L. Dallachiesa, **E. Börjeson**, R. Ryf, M. Bergroth, B. Josefsson, N. K. Fontaine, H. Chen, D. T. Neilson, J. Schröder, P. Larsson-Edefors, and M. Karlsson, "Real-Time Monitoring of Cable Break in a Live Network using a Coherent Transceiver Prototype", *Optical Fiber Communication Conference (OFC)*, p. Tu3J.6, Mar. 2024.

Contents

Acknowledgement

First, I would like to thank my supervisor, Prof. Per Larsson-Edefors, both for accepting me as a PhD student and for guiding me on this journey. The time and effort he has invested in me and my work have been invaluable. Our many discussions about topics other than research have been a large part of making this process as fun as it has been.

I would like to thank my co-supervisor, Prof. Magnus Karlsson for explaining the details of fiber-optic communication and fiber propagation, in a way much clearer than any textbook. I would also like to thank Assoc. Prof. Lars Svensson for the many discussions, both productive and not so productive, that we have had over the last few years.

I want to thank Dr. Mikael Mazur for inviting me to collaborate with him on transmission and fiber-sensing experiments, and for the interesting talks we have had about these and other systems. I extend my gratitude to my colleagues and office mates, Dr. Christoffer Fougstedt, Dr. Victor Åberg, Dr. Erik Ryman, Siavash Mowlavi and other more short-term acquaintances, for collaborations, discussions and for enduring sharing an office with me. I also want to thank Dr. Lars Lundberg for introducing me to this field of research during my master's thesis project.

There are people outside of my research field that also deserve to be thanked: Rasmus Blanck, for being an inspiration and a supporter throughout this process, for distractions and for proofreading and typesetting support. Tobias Kjerstadius, for always being available for a quick beer on the way home. Per Malm, for discussions of research, academia and everything else. To my dad, Björn Börjeson, for proofreading and for being a large part of making me a fairly well-adjusted person.

Finally, my deepest gratitude goes to my love, Sigrid. Her patience and support have made all the difference. Without her, this would have been infinitely harder.

Erik Börjeson
Göteborg, March 2024

Acronyms

ADC	analog-to-digital converter
ASIC	application-specific integrated circuit
AWGN	additive white Gaussian noise
BER	bit error rate
BPS	blind phase search
CC-MCF	coupled-core multi-core fiber
CD	chromatic dispersion
CLB	configurable logic block
CMA	constant-modulus algorithm
CMOS	complementary metal-oxide semiconductor
CORDIC	coordinate rotation digital computer
CPR	carrier phase recovery
CT	constellation transformation
DCF	dispersion-compensating fiber
DD	decision-directed
DGD	differential group delay
DSP	digital signal processing
EDFA	erbium-doped fiber amplifier
EVM	error-vector magnitude
FEC	forward error correction
FFT	fast Fourier transform
FIR	finite impulse response
FoC	Fiber-on-Chip
FPGA	field-programmable gate array
GVD	group-velocity dispersion
HDL	hardware description language
HLS	high-level synthesis
IFFT	inverse fast Fourier transform
IM/DD	intensity modulation/direct detection

ISI	inter-symbol interference
LMS	least mean squares
LO	local oscillator
MCF	multi-core fiber
MIMO	multiple-input multiple-output
MLE	maximum likelihood estimation
MMCM	multiplierless multiple-constant multiplication
MSB	most significant bit
NLSE	non-linear Schrödinger equation
OOK	on-off keying
PCPE	principal component-based phase estimation
PMD	polarization mode dispersion
PSK	phase-shift keying
QAM	quadrature amplitude modulation
QPSK	quadrature phase-shift keying
RC	raised-cosine
RDE	radius-directed equalizer
RMS	root mean square
RRC	root-raised-cosine
SDM	space-division multiplexing
SMF	single-mode fiber
SNR	signal-to-noise ratio
SOP	state of polarization
UC-MCF	uncoupled-core multi-core fiber
VLSI	very-large-scale integration
VV	Viterbi–Viterbi

To Alva, Anna and Asta

Chapter 1

Introduction

The digital signal processing (DSP) systems used in fiber-optic communication links are one of the enabling technologies for the interconnected world that we live in today, and these systems can be significant contributors to the overall power dissipation of a link [1]. The number of connected users and their bandwidth requirements are expected to continue to grow over the coming years [2]. To facilitate this growth, the development of power-efficient high-throughput DSP systems becomes increasingly important. To support higher data throughput, the complexity of the fiber-optic links grows when more of the physical dimensions available for modulation and multiplexing are used for transmission at faster rates [3]. This results in more complex DSP systems, which will have to be implemented under tight energy constraints and which will be more challenging to verify.

The vast majority of Internet traffic is carried over fiber-optic communication links, enabled by the invention of the laser [4], low-loss fibers [5, 6] and the erbium-doped fiber amplifier (EDFA) [7]. Today, long-haul links are dominated by coherent transmission over single-mode fibers (SMFs) [8]. As coherent systems allow capture of both amplitude and phase, the spectral efficiency of these systems is much better than traditional on-off keying (OOK). In this thesis, we will focus on intradyne coherent systems, which have free-running local oscillator (LO) lasers, as these have the additional advantage of simplifying the optical hardware by eliminating the optical phase-locked loop. However, these systems rely heavily on DSP, typically implemented as an application-specific integrated circuit (ASIC), to compensate for transmission impairments such as dispersion or phase noise.

Increasing the modulation order of the SMF systems is one way of supporting continuing bandwidth growth. Unfortunately, that will also increase the power dissipation of the DSP ASICs. Furthermore, when coherent links are starting to find their way also into short-haul systems, such as data-center interconnects, the energy efficiency aspects become even more important as these applications have a more limited power budget than long-haul, due to the large number of connections [9].

An alternative to improving SMF systems, and continuing the capacity growth, is space-division multiplexing (SDM) using e.g. coupled-core multi-core fiber (CC-MCF) [10]. This technique, although promising from an optical perspective, needs power-demanding multiple-input multiple-output (MIMO) processing in the DSP ASIC to deal with the coupling and differential group delay introduced during signal propagation.

For many years, circuit designers have been able to rely on chip foundries delivering faster circuits with lower power dissipation for each new process node. As feature scaling is coming to an end, due to both physical and economical factors [11], this might no longer be true, forcing algorithm and circuit designers to strive for more sophisticated and efficient solutions to increasingly complex problems.

In this thesis, we explore circuit implementations of DSP algorithms and study how design choices and parameter settings affect power dissipation, throughput and signal degradation. By uncovering the trade-off between these aspects, more informed system design decisions can be made. The thesis include publications related to carrier phase recovery (CPR) and dynamic MIMO equalization of CC-MCF systems. Additionally, we present a circuit verification and evaluation environment that can run on ASICs and field-programmable gate arrays (FPGAs). By emulating a complete fiber-optic transmission system in hardware, this enviroment can be used to e.g. reduce simulation runtimes compared to traditional approaches.

1.1 Thesis Outline

The remainder of this thesis will describe the properties of a coherent fiber-optic communication system in Chapter 2, focusing on how to model the relevant channel impairments. The relevant DSP subsystems and algorithms are presented in Chapter 3. Chapter 4 describes the hardware platforms commonly used to realize a coherent DSP, and the process of circuit design and evaluation. Finally, Chapter 5 summarizes the contributions of the included publications.

Chapter 2

Fiber-Optic Communication Systems

The purpose of a fiber-optic communication system is to transfer data between two points using an optical fiber as the communication medium. The data are typically digital and encoded to symbols in a transmitter. The symbols are used to modulate the output from a carrier laser, before launching the optical signal into the fiber. On the receiving end, the optical signal is converted to an electrical signal, sampled and demodulated back into binary data.

The simplest form of fiber-optic communication is based on encoding data on the amplitude of the carrier laser using intensity modulation/direct detection (IM/DD) and a modulation format such as OOK, where two amplitude levels represent 0 or 1. More amplitude levels can be added to increase the data throughput by encoding multiple bits per transmitted symbol. Using four amplitude levels (PAM4), the data rate can be doubled compared to OOK, assuming a constant symbol rate. The transmitter and receiver for IM/DD detection can be relatively simple, but they do not utilize the full optical field.

The introduction of coherent receivers for fiber-optic communication enables encoding and detection of both amplitude and phase-modulated data. By combining these two modulation methods, quadrature amplitude modulation (QAM) can be used, increasing the data throughput. The symbols can be viewed as points in the complex plane, where each symbol has a magnitude and a phase, and Fig. 2.1 shows constellation plots for four common modulation formats, where the axes represent the in-phase (I) and quadrature (Q) components of the signals. Additionally,

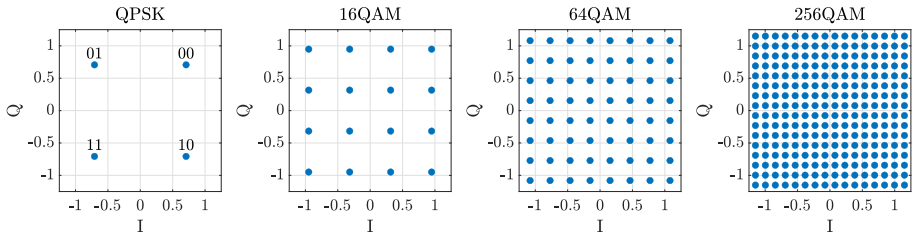

Figure 2.1: Constellation diagram of (a) QPSK, also showing the encoded bit values, (b) 16QAM, (c) 64QAM and (d) 256QAM.

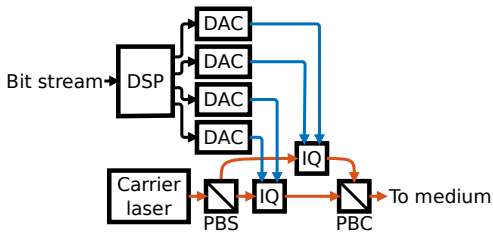

Figure 2.2: Simplified block diagram of a coherent transmitter, where black, blue and red arrows represent optical signals, digital electrical signals, and analog electrical signals, respectively.

Fig. 2.1a shows one way to encode two bits on quadrature phase-shift keying (QPSK) symbols. This example uses Gray encoding, where the closest adjacent symbols only differ by one bit to minimize bit errors, should the symbols be misclassified in the receiver. Note that if we assume a uniform symbol probability, the average energy for all modulation formats in Fig. 2.1 is equal.

2.1 Transmitter

A simplified block diagram of a dual polarization coherent transmitter for SMF is shown in Fig. 2.2, where the bit stream to be transmitted is first passed through a DSP stage. The operations performed in this stage vary between systems, but at the very minimum symbol encoding and pulse shaping are included. Additional processing steps can include forward error correction (FEC) encoding as well as precompensation of transmitter and channel impairments.

A system with square pulses in the time-domain has an infinite frequency response, thus pulse shaping of the encoded symbols is applied

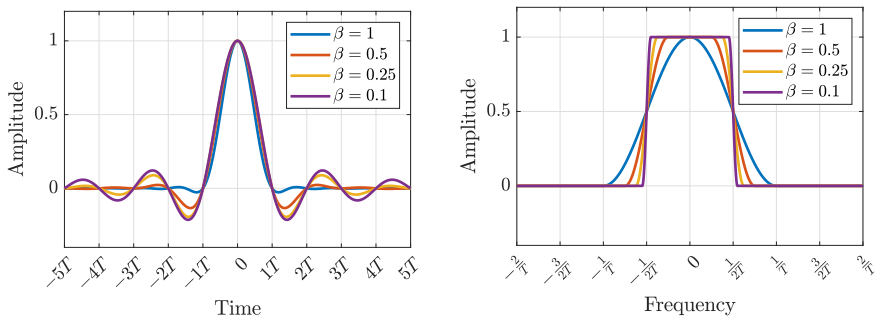

Figure 2.3: Raised cosine (a) impulse and (b) frequency response for different roll-off factors.

to limit the bandwidth of the signal. The ideal would be a rectangular spectra, as this enables dense packing of channels. However, such a frequency response corresponds to a sinc pulse in the time-domain, which has infinite duration. A trade-off is shaping the pulses using a raised-cosine (RC) filter

$$h(t) \begin{cases} \frac{\pi}{4T}\,\mathrm{sinc}\left(\frac{1}{2\beta}\right) & t = \pm\frac{T}{2\beta} \\ \frac{1}{T}\,\mathrm{sinc}\left(\frac{t}{T}\right)\frac{\cos\left(\frac{\pi\beta t}{T}\right)}{1-\left(\frac{2\beta t}{T}\right)^2} & \text{otherwise,} \end{cases} \tag{2.1}$$

where h is the filter response, T is the symbol period and β is the roll-off factor. The relation between the time and frequency responses of an RC filter is shown in Fig. 2.3, where a lower β results in a narrower frequency response. However, the corresponding impulse response is longer, which results in filters with more resource-demanding hardware implementations.

To maximize the signal-to-noise ratio (SNR) of the received signal, pulse shaping is usually implemented as a matched filter, where the filter is divided between the transmitter and the receiver. At each point, the square root of the RC filter response is used, a root-raised-cosine (RRC) filter, which combines into a RC filter response. Note that in order to perform pulse shaping of the signal, it needs to be upsampled.

IQ signals representing the transmitted data are fed from the DSP to digital-to-analog converters, which drive the IQ modulators. If the system utilizes both polarizations in an SMF, two modulators are necessary and the polarizations are combined in a polarization beam combiner before being launched into the fiber.

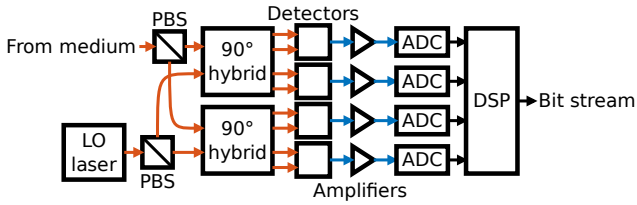

Figure 2.4: Simplified block diagram of a coherent receiver, where black, blue and red arrows represent optical signals, digital electrical signals, and analog electrical signals, respectively.

2.2 Receiver

A block diagram of a coherent receiver is shown in Fig. 2.4, for a dual polarization system. In the first stage, the optical signal is split into two orthogonal polarization states using a polarization beam splitter. The resulting signals are mixed with the LO laser in 90°-hybrids before being converted to electrical signals using photo detectors. After amplification, these signals, which represent the I and Q components of the two polarizations, are fed to analog-to-digital converters (ADCs). Postprocessing to recover the transmitted bitstream is performed using a DSP system, which is the topic of Chapter 3.

2.3 Fiber Channel

There are many types of optical fibers, but this work focuses mainly on systems employing SMF in papers I–IV and multi-core fiber (MCF) in paper V. The SMF consists of a single core designed to support a single fundamental mode and transmission over two polarizations is possible. MCFs contain multiple cores in the same cladding, offering the possibility to transmit multiple signals simultaneously, thus increasing the data throughput by space-division multiplexing (SDM). The MCF type can be further subdivided into coupled-core (CC-MCF) and uncoupled-core (UC-MCF). To realize an UC-MCF, the core-to-core distance needs to be larger than for CC-MCF, to avoid crosstalk between the different cores. In a CC-MCF the core-to-core distance is chosen to maximize the coupling, which results in a slower accumulation of differential group delay between the different cores [12]. The idea is to reduce the signal-processing complexity for the cross-talk compensation, as the impulse response becomes shorter.

Propagation of a single polarization in an SMF can be described using the non-linear Schrödinger equation (NLSE),

$$\frac{\partial A}{\partial z} = -\frac{j\beta_2}{2}\frac{\partial^2 A}{\partial t^2} - \frac{\alpha}{2}A + j\gamma|A|^2 A, \tag{2.2}$$

where A is the amplitude of the optical signal, α is the attenuation, β_2 is the group-velocity dispersion (GVD) parameter and γ is a parameter describing the fiber non-linearity [13]. For short-haul fiber-optic systems, the transmitter can be directly connected to the receiver, without additional amplification. However, fiber attenuation limits the possible reach for non-amplified systems. The attenuation of a fiber is wavelength dependent and SMFs have a minimum of approximately 0.2 dB/km at $\lambda = 1550$ nm [13]. Transmission in the C-band[1] (1530–1565 nm) is therefore often preferred. Despite this low attenuation, amplification is necessary for long-haul systems. EDFAs are commonly used and the fiber is divided into multiple spans, with amplifers inserted between them. These amplifiers are significant noise contributors in long-haul systems due to amplified spontaneous emission. The refractive index of a fiber is dependent on the optical power propagating through the fiber, as illustrated by the non-linear term in the NLSE (2.2). Thus, the issues associated with compensating for this nonlinearity limit the maximum power that can be launched into the fiber.

Transmission impairments do not only originate from signal propagation through the fiber. Optical and electrical amplifiers, electronic components etc. can also contribute to signal degradation. The following sections will describe the transmission impairments most relevant to this work in more detail.

2.3.1 Additive White Gaussian Noise

As discussed above, EDFAs are significant noise contributors for long-haul systems and are often the dominant noise source. The electronic parts of the transmitter and receiver further increase noise due to shot noise from the photo diodes, which can be a significant for short-haul systems, and thermal noise from e.g. electrical amplifiers.

Usually, these noise sources are modeled using an additive white Gaussian noise (AWGN) channel model

$$y[n] = x[n] + w[n], \tag{2.3}$$

[1]The spectrum used for fiber-optic communication is divided into a number of frequency bands: O, E, S, C, L and U, where C stands for conventional.

where x is the transmitted symbol, y is the received symbol and n is the sample index. The noise term, w, is a Gaussian distributed random variable with zero mean and a variance of N_0 [14].

The channel capacity (C), i.e. the theoretical maximum reliable throughput in bits/s/Hz, of an AWGN channel can be calculated as

$$C = \log_2\left(1 + \frac{S}{N_0}\right), \tag{2.4}$$

where S is the expected value of the signal power [15]. This property makes the AWGN channel useful as a reference when developing algorithms and circuit implementations of DSP systems.

2.3.2 Chromatic Dispersion

The propagation speed of optical pulses in a fiber is wavelength dependent, and can be described by the GVD parameter β_2 in the NLSE (2.2). As different spectral components of a pulse travel at different speeds, the result is pulse broadening, or chromatic dispersion (CD), which can be described by the transfer function

$$G(z, \omega) = \exp\left(j\frac{\omega^2 \beta_2 z}{2}\right), \tag{2.5}$$

where ω is the angular frequency and z is the propagation distance [13]. The GVD is related to the dispersion parameter D, typically expressed in ps/(km·nm), as

$$D = -\frac{2\pi c}{\lambda^2}\beta_2, \tag{2.6}$$

where c is the speed of light and λ is the wavelength. Dispersion characteristics are related to the fiber medium, but can be controlled using dopants and by adjusting waveguide properties. This possibility enables construction of e.g. dispersion-compensating fibers (DCFs), which have a dispersion parameter with an opposite sign to that of a standard fiber.

For long-haul fibers, the dispersion can become a severe problem, as pulses start to overlap and cause inter-symbol interference (ISI). The ISI must be compensated before the data can be properly recovered by the receiver. Compensation can be performed by inserting a DCF in the link. However, these fibers have higher loss and larger issues with non-linearities than a standard SMF. The CD compensation can also be handled in DSP using methods described in Sec. 3.1.

2.3.3 Polarization Effects

An SMF fiber can support propagation over two orthogonal polarizations. In an ideal, perfectly cylindrical fiber, no coupling will occur between these two polarizations. In reality, small variations exist in the fiber cross section and the fiber is exposed to external stresses, such as bends and vibrations. These factors cause fiber birefringence, which varies randomly over the fiber length. As a result, the two signals will couple randomly during propagation as the state of polarization (SOP) change. They will also exhibit different propagation speeds, resulting in polarization mode dispersion (PMD), where the differential group delay (DGD) is the time delay between the two polarizations [13]. Usually, the SOP drifts slowly, but fast polarization changes can occur as isolated events due to external influences [16]. It is worth noting that the pulse broadening caused by PMD is relatively small compared to the effects of CD [13].

The polarization effects can be modeled as a concatenation of multiple fiber segments with static birefringence, and thus a constant DGD and stable SOP,

$$\boldsymbol{H}_k = \begin{bmatrix} e^{j\omega\tau_k/2} & 0 \\ 0 & e^{-j\omega\tau_k/2} \end{bmatrix}, \tag{2.7}$$

where τ_k is the DGD of the kth segment. These segments are connected with phase and polarization rotators, having zero DGD, described by the Jones matrices

$$\boldsymbol{D}_k = \begin{bmatrix} e^{-j\phi_k} & 0 \\ 0 & e^{j\phi_k} \end{bmatrix} \begin{bmatrix} \cos\theta_k & \sin\theta_k \\ -\sin\theta_k & \cos\theta_k \end{bmatrix}, \tag{2.8}$$

where ϕ_k and θ_k are random rotation angles. This model can be further extended with time and frequency dependent elements, and more elaborate descriptions of $\boldsymbol{D}$ [17, 18]. An illustration of how a 16QAM transmission is affected by a single polarization rotator is shown in Fig. 2.5.

2.3.4 Phase Noise

The carrier and LO lasers used in the transmitter and receiver of a coherent system are not completely stable, but their phases vary with time. The coherence time of a laser is defined as the time over which it is coherent, i.e. has a stable phase. This time should be much longer than the symbol duration [19]. A common property used to describe lasers is the linewidth, Δv, which is inversely related to the coherence time.

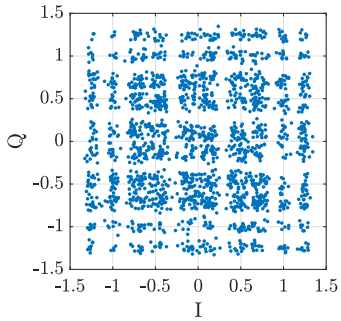

Figure 2.5: Illustration of how an SOP rotation with $\theta = \pi/8$ affects one of the polarizations in a dual polarization transmission of 2000 16QAM symbols.

If two lasers are used, such as in an intradyne coherent system, the sum of their linewidths can be used as a measurement of the total system linewidth. Since the effect of phase noise on the transmitted symbols is dependent on the symbol duration T_s the dimensionless metric $\Delta v T_s$ is often used to quantify the phase noise impact.

Phase noise can be modeled as a Wiener process

$$\phi_k = \phi_{k-1} + \Delta\phi_k, \tag{2.9}$$

where ϕ_k is the phase of the kth symbol and $\Delta\phi_k$ is a Gaussian-distributed random variable with zero mean and a variance

$$\sigma^2_{\Delta\phi_k} = 2\pi\Delta v T_s, \tag{2.10}$$

where Δv is the combined linewidth of the carrier and LO lasers [20]. Typical linewidths of lasers used in fiber-optic communication are in the order of 100 kHz, as lasers with narrower bandwidths comes at a much higher cost. An example of how the phase noise affects a transmission is shown in Fig. 2.6, for two different $\Delta v T_s$, corresponding to a 10-GBaud transmission and laser linewidths of 100 kHz and 1 MHz.

For systems that perform CD compensation in DSP, equalization-enhanced phase noise (EEPN) can become an issue at high symbol rates and large accumulated CD. EEPN is caused by passing the LO laser signal through the CD compensation without it being subject to dispersive propagation through the fiber [21]. The result is an additional amplitude noise at the output of the CD compensation with the variance

$$\sigma^2_{EEPN} = \frac{\pi c D L \Delta v_{LO}}{2 T_s f_o^2}, \tag{2.11}$$

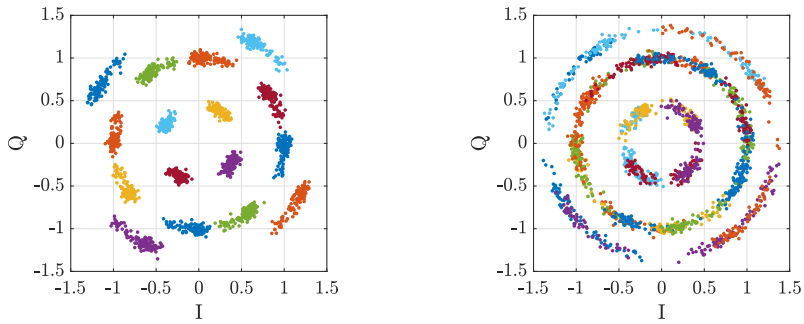

Figure 2.6: Illustration of how phase noise affects a 10 GBaud transmission of 2000 16QAM symbols, for (a) $\Delta v T_S = 10^{-5}$ ($\Delta v = 100$ kHz) and (b) $\Delta v T_s = 10^{-6}$ ($\Delta v = 1$ MHz).

where L is the fiber length and f_0 is the center frequency of the LO laser [21]. This additional noise puts more stringent requirements on LO laser linewidth and system SNR. However, the noise caused by EEPN is not Gaussian, but has a temporal correlation. Thus, if a Gaussian distribution is assumed, the impact of EEPN can be overestimated when a CPR subsystem is employed in the DSP [22]. In this work, we do not consider the effects of EEPN on our CPR implementations, as it is better compensated for using other methods such as digital coherence enhancement [23] or electrical feedback to the LO laser [24].

Chapter 3

Signal Processing Subsystems

The goal of the digital signal processing system employed in coherent fiber-optic communication systems is to reconstruct the transmitted data stream from the received signal. The DSP is typically realized as a chain of subsystems, where each part is designed to compensate for a subset of the impairments described in Chapter 2.

There are many different realizations of coherent receivers, but a typical layout is shown in Fig. 3.1. The first stage compensates for any imbalances in the optical front-end, such as imperfections in the 90° hybrids or mismatched photo-detector response, and is followed by linear equalization. In principle, all linear impairments could be compensated by the same equalizer. However, the static impairments typically have a longer memory than the dynamic. By splitting equalization into two stages, a longer static equalizer can be used to compensate for the static impairments, while the filter length of the more complex dynamic equalizer can be reduced [25].

The sample clocks of the transmitter and receiver are usually not synchronized, which means that timing recovery is necessary. Usually,

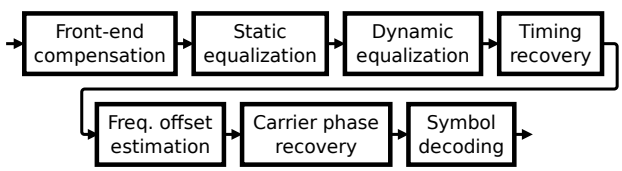

Figure 3.1: Typical architecture of a coherent DSP.

the receiver employs oversampling of the input signals in the ADCs and the timing recovery subsystem is used to find the best sampling instance. Classical methods include Gardner [26] and Mueller and Muller [27].

If the carrier and LO lasers are not frequency locked, which is the case for intradyne systems, there may be a frequency offset between them. This offset will manifest itself as a constant phase rotation of the received symbols. To reduce the load on the carrier phase recovery subsystem, the frequency offset can be separately compensated using spectral methods [28], or using techniques similar to the 4th power estimator described in Section 3.3.3 [29]. Before decoding the symbols, carrier phase recovery is performed to compensate for the phase noise.

The following sections will describe solutions for equalization and carrier phase estimation subsystems in greater detail, since these are the main topics of this thesis.

3.1 Static Equalization

To compensate for CD and other static impairments, a static equalizer is typically used. The equalization can be realized using an all-pass filter with the transfer function

$$H_c(\omega) = 1/G(z,\omega) = G(-z,\omega), \tag{3.1}$$

where G is given in (2.5) [19]. The compensation can be performed either in the time or the frequency domain. The latter is preferred for longer systems, where the number of filter taps can otherwise become prohibitively large due to the long impulse response. For a frequency-domain implementation, the input signal first needs to be Fourier transformed before multiplication with the transfer function H_c, followed by an inverse Fourier transform. In a circuit implementation, fast Fourier transforms (FFTs) and inverse fast Fourier transforms (IFFTs) are usually used, employing the overlap-save or overlap-add methods to facilitate real-time implementation and avoid time-domain aliasing problems [30].

Time-domain solutions can be of interest for shorter links, where the complexity of the FFTs and IFFTs otherwise dominate. However, the impulse response of H_c is of infinite duration, and needs to be truncated to be able to employ a finite impulse response (FIR) filter [31]. Alternatively infinite impulse response filters could be used, but they tend to be harder to parallelize [25].

The matched filter can also be included in the static equalization, which might be necessary for RRC filters with β close to 0, to reduce the number of filter taps necessary for the dynamic equalization described next. The transfer function then becomes

$$H(\omega) = H_{RRC}(\omega)H_c(\omega). \tag{3.2}$$

However, if there is a large frequency offset between the carrier and LO lasers, a static matched filter comes with a penalty [32].

3.2 Dynamic Equalization

To handle linear time-dependent impairments, static equalization is often followed by a dynamic, adaptive filter. These filters are mainly used to compensate for polarization-dependent effects, such as SOP rotations and PMD. Typically, the memory-length of the dynamic effects is relatively small, which means that using time-domain equalizers can be a valid choice. For some specialized systems with longer impulse responses, frequency-domain filtering can be a more efficient choice [25]. Here, the convolution operations used in the time-domain filters are replaced by multiplications in the frequency-domain. For longer filters, this simplification outweighs the additional complexity of the FFTs and IFFTs neccessary for the transformation. In this work we focus on time-domain implementations.

For a dual polarization system, 2×2 MIMO FIR filters can be used to compensate for polarization effects by estimating the inverse Jones matrix. If we assume one sample per symbol and N taps per filter, the vectors containing the filter inputs, $\boldsymbol{x}_X$ and $\boldsymbol{x}_Y$, are

$$\begin{aligned} \boldsymbol{x}_X[n] &= \Big[x_X[n], x_X[n-1], ..., x_X[n-N+1]\Big]^T \\ \boldsymbol{x}_Y[n] &= \Big[x_Y[n], x_Y[n-1], ..., x_Y[n-N+1]\Big]^T, \end{aligned} \tag{3.3}$$

where n is the sample index and T is the transpose of a vector. The outputs y_X and y_Y are calculated as

$$\begin{aligned} y_X[n] &= \boldsymbol{h}_{XX}^H[n]\boldsymbol{x}_X[n] + \boldsymbol{h}_{XY}^H[n]\boldsymbol{x}_Y[n] \\ y_Y[n] &= \boldsymbol{h}_{YX}^H[n]\boldsymbol{x}_X[n] + \boldsymbol{h}_{YY}^H[n]\boldsymbol{x}_Y[n], \end{aligned} \tag{3.4}$$

where $\boldsymbol{h}_{ij}$ are column vectors containing the, often complex-valued, filter coefficients and H is the Hermitian transpose. However, this MIMO

architecture cannot compensate for I/Q imbalances, due to the joint filtering of the I/Q components. In such cases, 4×4 real-valued filters can be used instead [33].

The number of filter taps used in the equalizer depends on the amount of PMD present in the system, and can also be affected by matched filtering requirements. In the context of the latter, the filter can be shifted in time, which will introduce a delay to maintain causality. If the only goal of the equalizer is to compensate for SOP rotations, no memory is needed in the filters, i.e. each $\boldsymbol{h}$ can consist of a single complex coefficient.

The filter taps are continuously updated to track the time-varying impairments. For equalizers based on the least mean squares (LMS) method, we calculate errors compared to a target symbol d

$$\begin{aligned} e_X[n] &= d_X[n] - y_X[n] \\ e_Y[n] &= d_Y[n] - y_Y[n]. \end{aligned} \tag{3.5}$$

The filter coefficients used in (3.4) can be updated iteratively

$$\begin{aligned} \boldsymbol{h}_{XX}[n+1] &= \boldsymbol{h}_{XX}[n] + \mu \boldsymbol{x}_X[n] e_X^*[n], \\ \boldsymbol{h}_{XY}[n+1] &= \boldsymbol{h}_{XY}[n] + \mu \boldsymbol{x}_Y[n] e_X^*[n], \\ \boldsymbol{h}_{YX}[n+1] &= \boldsymbol{h}_{YX}[n] + \mu \boldsymbol{x}_X[n] e_Y^*[n], \\ \boldsymbol{h}_{YY}[n+1] &= \boldsymbol{h}_{YY}[n] + \mu \boldsymbol{x}_Y[n] e_Y^*[n], \end{aligned} \tag{3.6}$$

where μ is a constant used to control the step size, or scaling of each update, and e^* is the complex conjugate of the error [34]. Unfortunately, the target symbol is not usually directly accessible, so to apply LMS we will either need to use a symbol decision or known pilot symbols. The same type of equalizer can be extended to work also with e.g. coupled-core multi-core fiber transmission, by increasing the dimensionality of the MIMO [35].

In the following sections we will describe some of the most used methods for dynamic equalization in fiber-optic systems, including decision-directed and pilot-based LMS. The non-LMS methods described below can be modified to use (3.6) to perform the tap updates.

3.2.1 Constant-Modulus Algorithm

The constant-modulus algorithm (CMA) is a blind method of updating the equalizer coefficients. By forcing the amplitude of the signals to

a reference value, it can compensate for e.g. PMD effects [36]. CMA works independently of the phase offset and phase noise, since these impairments have no effect on the modulus of the received signal. As the same reference value is used for all input symbols, CMA works best for modulation formats with constant-modulus, such as QPSK. It can still converge when using multi-modulus formats, e.g. QAM, if the reference value is chosen carefully [37]. However, the solution will not be optimal as the CMA error does not go to zero for a perfectly equalized signal. Instead it will converge to some average location, which is a trade-off between possible moduli.

For each polarization, the error is calculated as

$$e[n] = R - |y[n]|^2, \tag{3.7}$$

where R is a constant depending on the target symbol [38]:

$$R = \frac{\mathbb{E}\left[|d\left[n\right]|^4\right]}{\mathbb{E}\left[|d\left[n\right]|^2\right]}. \tag{3.8}$$

To be able to use the same tap-update function as LMS (3.6), the error can be rewritten as

$$e[n] = y[n](R - |y[n]|^2). \tag{3.9}$$

For systems using multi-modulus modulation formats, CMA can be used for initial convergence after which the equalizer is switched to a decision-directed mode, described in Section 3.2.3. Formulating the error function as (3.9) simplifies a circuit description of such an equalizer.

A drawback with the CMA equalizer is that the outputs can converge to the same polarization. Several methods have been described to avoid this behavior, e.g. by using a known preamble and pilot-based LMS to initialize the filters [39]. Another alternative is to initially only update the filters related to one of the inputs (e.g. $\boldsymbol{h}_{XX}$ and $\boldsymbol{h}_{YX}$). Once these have converged, they can be used to initialize the remaining filters,

$$\begin{aligned} h_{YY}[n] &= h^*_{XX}[-n] \\ h_{XY}[n] &= h^*_{YX}[-n], \end{aligned} \tag{3.10}$$

where * is the complex conjugate [40]. The CMA update can then progress as described above. However, this solution assumes that the two polarization inputs are orthogonal to each other, which is not always the case.

3.2.2 Radius-Directed Equalization

To improve the performance of CMA for signals with multiple moduli, multiple values of the R constant can be used for different received symbols. In a radius-directed equalizer (RDE), the error between the equalizer output modulus and the nearest symbol radius is calculated as

$$e[n] = y[n]\left(Q\left(y[n]\right)^2 - |y[n]|^2\right), \tag{3.11}$$

where Q is a function performing a hard decision of the radius of its input, based on a set of reference radii [41]. The error is used as input to the tap update (3.6) in the same way as for CMA.

The RDE is sensitive to errors in the hard decision, and can thus have trouble converging initially, when symbol errors are frequent. A solution is to employ CMA to preconverge the filters, before switching to RDE. Performance on dynamic channels can be improved further for formats beyond 16QAM by scaling the error function with the relative probability for each received radius [42].

3.2.3 Decision-Directed Least Mean Square

Another type of hard decision is performed in the decision-directed (DD)-LMS equalizer. If we replace d in (3.5) with a symbol decision of the equalizer output $\hat{y}$, i.e. the constellation point closest to y, we get a new error function [43]:

$$e[n] = \hat{y}[n] - y[n]. \tag{3.12}$$

As this is a hard-decision algorithm, its performance depends heavily on the symbol error probability. Thus, pre-convergence using e.g. CMA might be necessary, similarly to RDE.

To perform the hard decision in a realistic DSP system, the equalizer needs to be followed by carrier phase recovery (Section 3.3) before decoding. The latency of the feedback caused by these additional blocks will affect the convergence and tracking performance of the DD equalizer, especially when considering a very dynamic channel [44].

3.2.4 Pilot-Based Least Mean Squares

Instead of using symbol decisions, known pilot symbols time-division multiplexed into the data-symbol stream can be used as d in the LMS error function (3.5) [45]. However, this typically means that we need to perform less frequent updates of the filter coefficient, which results in a

slower convergence speed and a higher sensitivity to fast SOP changes. The former can be mitigated by starting the transmission with a preamble consisting of a longer group of pilot symbols. The pilot symbols can be of a simple phase-shift keying (PSK) modulation format, independent of the modulation format used for the data carrying symbols.

3.3 Carrier Phase Recovery

The CPR algorithms should compensate for the phase noise present in the received signal. The process can usually be divided into two stages, where the phase noise is first estimated using one of the methods described below. The estimated phase is then used to compensate for the phase noise as

$$y[n] = x[n]e^{-j\hat{\phi}[n]}, \tag{3.13}$$

where x and y are the input and output symbols of the CPR, respectively, and $\hat{\phi}[n]$ is the estimated phase of the nth input. The differences between the algorithms presented below are usually related to the estimation part, while the compensation is often implemented as in (3.13).

CPR can be performed using either data-aided or blind approaches. In the former, known pilot symbols are inserted in the data stream and used for estimation, while the latter uses the received data-carrying symbols to estimate the phase. Blind estimation is relatively straightforward for simple modulations formats, such as M-PSK, where we can use the Viterbi–Viterbi (VV) estimator. For more complex formats, such as M-QAM, more elaborate algorithms are needed because of the multiple possible magnitudes of the symbols. Additionally, higher-order formats are more sensitive to phase noise as the Euclidean distance between constellation points is smaller [46].

To improve the phase-noise estimation, multi-stage estimators have been suggested, where a first stage performs a coarse estimation, which is refined by the subsequent estimation stages. Multi-stage CPR can be implemented using the same method for all stages [47], or by mixing different methods [48–51].

3.3.1 Cycle Slips

Most modulation formats considered for fiber-optic transmission have a $\pi/2$ symmetry of the constellation points. Additionally, the phase estimation often has a limited range. In most CPR implementations this

limitation is handled by including an unwrapping operation as the final stage of the estimator. For an estimator with a range of the estimated phase between 0 and $\pi/2$, unwrapping can be performed as

$$\hat{\phi}[n] \leftarrow \hat{\phi}[n] + \left\lfloor \frac{1}{2} + \frac{\hat{\phi}[n-1] - \hat{\phi}[n]}{\pi/2} \right\rfloor \frac{\pi}{2}, \tag{3.14}$$

where $\lfloor \cdot \rfloor$ denotes the floor function [52]. When the estimated phase reaches the end of the range and wraps around to the other end, these large phase jumps are detected and handled. A cycle slip occurs if this jump was actually present in the input signal or if the unwrapping operation fails to detect a wrap-around. In such cases the estimated phase will have an error of a multiple of $\pi/2$ and none of the following symbols will be correctly decoded.

The cycle-slip probability differs between CPR methods and increases with a larger $\Delta f T_s$, lower SNR and a shorter estimation window [53]. An acceptable cycle-slip probability for a production system can be as low as 10^{-18} [54]. Differential encoding of the data can be used to mitigate the effects of cycle slips, but causes a pre-FEC bit error rate (BER) increase [53]. The cycle-slip probability can also be reduced using blind methods [55, 56]. Alternatively, pilot symbols can used to recover from cycle-slips [53].

Analytical analysis of the cycle-slip probability is complicated for feed-forward systems [57], and analyzing circuit implementations of these systems is even more challenging. Using systems such as FoC, described in Section 4.4, which facilitates the processing of large number of symbols, can make estimation of the cycle-slip probability feasible.

3.3.2 Pilot-Symbol Aided Phase Recovery

Pilot-based CPR algorithms estimate the phase noise using known symbols, which are time-division multiplexed with the data symbols. These pilot symbols can be of a simple modulation format, such as QPSK, and estimation of the phase of the kth pilot symbol can be performed as

$$\hat{\phi}[k] = \arg(p[k]q^*[k]), \tag{3.15}$$

where p is the received pilot symbol and q^* is the complex conjugate of the expected symbol. In practice, a filter is used before calculating the argument to interpolate the phase between the pilots [58], and additional averaging over multiple pilot symbols can be necessary at low SNRs.

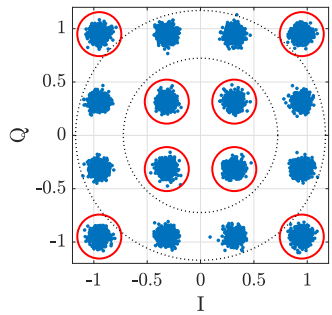

Figure 3.2: 16QAM constellation with the Class-1 symbols circled and the amplitude boundaries marked with dotted lines.

To track the phase for larger $\Delta f T_s$, a higher pilot-insertion ratio is needed, reducing the data throughput. However, a pilot-aided CPR can be followed by a blind stage to lower the necessary pilot overhead [59].

3.3.3 Viterbi–Viterbi

For M-PSK systems, the Viterbi–Viterbi (VV), or Mth-power, phase estimator can be used [60]. It removes the phase modulation by taking the Mth power of the input symbols as

$$\hat{\phi}[n] = \arg(x^M[n])/M, \tag{3.16}$$

In practice, a moving average of spanning $N = 2L + 1$ symbols

$$a[n] = \frac{1}{N} \sum_{l=-L}^{L} x^M[n+l], \tag{3.17}$$

is usually applied before calculating the argument, to reduce the impact of AWGN.

The VV estimator performs well for PSK modulation formats, but breaks down for higher-order formats, such as QAM, as these formats encode data also on the magnitude of the symbols. However, much effort has been put into modifying the algorithm to work also with these formats, mainly due to the simplicity of the original solution. One such approach is QPSK partitioning [61], where QAM symbols are split into Class-1 symbols, having a modulation angle of $\pi/4 + n\pi/2$ for $n = 0 \ldots 3$, and Class-2 symbols having other modulation angles. The partitioning can be performed by studying the magnitudes of the received symbols, as shown in Fig. 3.2, where the Class-1 symbols are circled in red and the

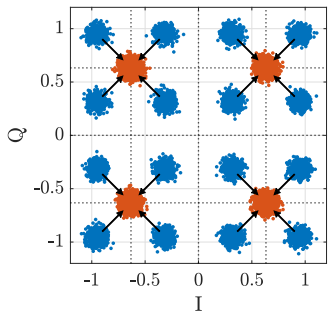

Figure 3.3: Constellation transformation of 16QAM to QPSK with the decision boundaries marked with dotted lines.

classification limits are marked with dashed lines. The Class-1 symbols are used to perform VV estimation after normalization of the amplitudes. Optionally, the symbols can be given different weights in the averaging, to account for the fact that the inner symbols have a lower SNR [62].

A drawback of this approach is that only a fraction of the symbols are used in the estimation, especially for constellations larger than 16QAM. As fewer symbols are used, a longer averaging window is needed, resulting in reduced performance for larger $\Delta v T_s$ values. It is possible to include also the symbols in the middle ring in Fig. 3.2, as these can be considered two separate QPSK constellations rotated $\pm(\pi/4 - \arctan(1/3))$ [63]. This will reduce the required averaging window length, but comes at the additional cost of correctly removing the modulation phase offset from these symbols and distinguishing between the two QPSK constellations.

For fine-grain CPR, used in the final stages of multi-stage CPR solutions, other methods can be used to facilitate the use of VV for higher-order formats. In [64], a method called constellation transformation (CT) is introduced, which transform 16QAM signals to QPSK. Assuming that the 16QAM constellation points are located at $\{\pm 1 \pm i, \pm 1 \pm 3i, \pm 3 \pm i, \pm 3 \pm 3i\}$ the transformation can be expressed as

$$Y = \Big(X_I - \mathrm{sgn}\big(X_I - 2\,\mathrm{sgn}(X_I)\big)\Big) + i\Big(X_Q - \mathrm{sgn}\big(X_Q - 2\,\mathrm{sgn}(X_Q)\big)\Big), \quad (3.18)$$

where X_I and X_Q are the I and Q components of the 16QAM symbol, Y is the resulting QPSK symbol and $\mathrm{sgn}(\cdot)$ is the signum function. For this method to perform well, the residual phase noise left from previous CPR stages must be relatively small, as it needs to perform a type of symbol decision in the transformation. A graphical illustration is shown in Fig. 3.3, where the decision boundaries are marked with dashed lines.

A description of how to extend CT to work also with 64QAM, by stepwise transformations of 64QAM to 16QAM to QPSK, is presented in [65].

3.3.4 Blind Phase Search

Blind phase search (BPS) is another example of a blind CPR algorithm [66]. Compared to the previously discussed methods, it is a more brute-force approach where a range of test angles is used to rotate the received symbols in the complex plane, and the symbols rotated by the angle resulting in the minimum average distance to a constellation point are chosen as the output. The B test angles are selected in the range 0 to $\pi/2$ as

$$\theta_b = \frac{b}{B} \cdot \frac{\pi}{2} \qquad b = 0, 1, ..., B-1. \tag{3.19}$$

These angles are used to rotate each input symbol as

$$x_b[n] = x[n]e^{j\theta_b}. \tag{3.20}$$

For each of the resulting rotated copies of the input symbol, the squared distance $d_b[n]$ to its closest constellation point $\hat{x}_b[n]$ is calculated by

$$|d_b[n]|^2 = |x_b[n] - \hat{x}_b[n]|^2. \tag{3.21}$$

To suppress noise, the sum of squared distances for each test angle is calculated using a sliding window of $N = 2L + 1$ samples:

$$s_b[n] = \sum_{l=-L}^{L} |d_b[n+l]|^2. \tag{3.22}$$

The output symbol is selected by minimizing $s_b[n]$ over the test angles. If differential encoding of the quadrant bits is used, these can be used to detect quadrant jumps. Otherwise, the algorithm needs to track the current quadrant of the estimated phase using, e.g., the method described in (3.14). However, separate phase compensation is not necessary as the phase rotations have already been performed in (3.20).

The two main parameters controlling BPS behavior are the number of test angles, B, and the length of the window, N. The optimal settings are dependent on the modulation format, $\Delta v T_s$ and the SNR of the input symbols. Unfortunately, this dependency means that the complexity of a BPS implementation becomes very large for higher-order modulation formats, which need a higher test-angle resolution. The problem is

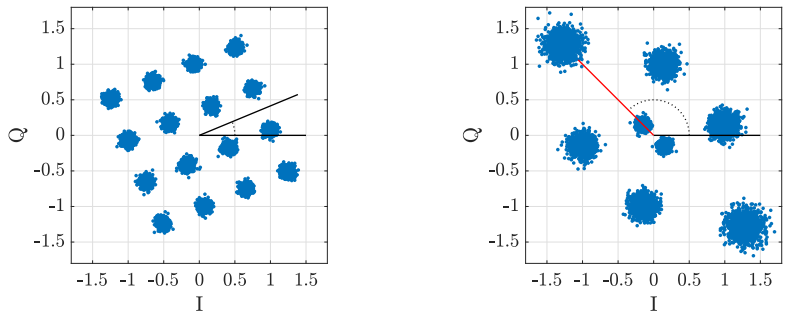

Figure 3.4: Assuming 16QAM input symbols to the principal component phase estimator, shown in (a), (b) shows the squared input symbols, with the principal component marked with a red line

exacerbated if the implementation needs to be parallelized for higher throughput.

In order to decrease the number of test angles, and keep the complexity under control, parabolic interpolation of the estimated phase using the average distances s_b has been suggested [67]. Assuming that $b = m$ results in the minimum s_b, the interpolation can be performed by calculating a new estimated phase by

$$\hat{\phi}[n] = \theta_m + \frac{\Delta\theta}{2} \cdot \frac{s_{m-1}[n] - s_{m+1}[n]}{s_{m-1}[n] + s_{m+1}[n] - 2s_m[n]}, \tag{3.23}$$

where $\Delta\theta$ is the difference of two adjacent test angles. Since rotation with this new estimated phase has not been performed in (3.20), a separate compensation operation, as in (3.13), is now needed. However, the total complexity is still reduced as the method can reduce the number of test phases by at least 75% [67].

Another way to reduce the implementation complexity of BPS is to split the estimation into multiple stages [47]. If B test angles are used in a one-stage approach, typically $\sqrt{B}$ angles are needed for each stage in a two-stage BPS to reach the same resolution. However, care must be taken to optimize the averaging length L of each stage to reach a similar performance as for a one-stage BPS.

3.3.5 Principal Component Phase Estimation

Principal component-based phase estimation (PCPE), presented in [68], estimates the phase of a block of input symbols using principal component

analysis [69]. Fig. 3.4a shows a block of 16QAM symbols with a $\phi = \pi/8$ phase rotation. Squaring these symbols results in the constellation shown in Fig. 3.4b, where the principal component (PC) is marked with a red line. The PC will have a phase angle of $2\phi + \pi/2$.

To calculate the principal component, the squares of N successive input symbols are calculated by

$$\boldsymbol{a}_l = \left[x^2[n] \;\; x^2[n-1] \;\; ... \;\; x^2[n-N+1],\right], \tag{3.24}$$

for the lth block of symbols. The $2 \times N$ matrix

$$\boldsymbol{A}_l = \begin{bmatrix} \Re(\boldsymbol{a}[1]) & \Re(\boldsymbol{a}[2]) & ... & \Re(\boldsymbol{a}[N]) \\ \Im(\boldsymbol{a}[1]) & \Im(\boldsymbol{a}[2]) & ... & \Im(\boldsymbol{a}[N]) \end{bmatrix}, \tag{3.25}$$

can then be used to calculate a 2×2 covariance matrix:

$$\boldsymbol{C}_l = \boldsymbol{A}_l \boldsymbol{A}_l^\top. \tag{3.26}$$

If we set the first PC to $\boldsymbol{v_0} = \begin{bmatrix} 1 \, 0 \end{bmatrix}^\top$, it can then be updated for each new block by

$$\boldsymbol{v}_l = \boldsymbol{C}_l \boldsymbol{v}_{l-1}. \tag{3.27}$$

An estimation of the phase rotation is given by

$$\hat{\phi}_l = \frac{1}{2} \arctan\left(\frac{\boldsymbol{v}_l[2]}{\boldsymbol{v}_l[1]}\right) - \frac{\pi}{4}, \tag{3.28}$$

which needs to be followed by an unwrapping operation, described in Section 3.3.1. The resulting phase estimation can be used to compensate the phase rotation for all symbols in the input block.

The block length N should be long enough to suppress noise, but for higher $\Delta v T_s$ a long window will result in reduced performance. Compared to BPS, Diniz et al. [68] show that PCPE has a lower cycle-slip probability but a worse BER performance at higher SNRs. To take advantage of the former, they suggest a two-stage approach where PCPE is used as a coarse phase estimator and a simplified BPS is used as a second stage. Since a fine-grained BPS can be implemented without phase unwrapping, it will not affect the cycle-slip probability.

3.3.6 Maximum Likelihood Phase Estimation

Maximum likelihood estimation (MLE) of the phase has been suggested as a final stage of multi-stage CPR approaches [50, 62]. By making a

decision $\hat{y}$ as the constellation point closest to the input symbol, the residual phase can be estimated in a way similar to the pilot-based approach, using a moving average to suppress noise,

$$a[n] = \sum_{l=-L}^{L} x[n+l]\hat{y}[n+l]^*. \tag{3.29}$$

The phase can then be extracted by

$$\hat{\phi}[n] = \arg(a[n]), \tag{3.30}$$

and used to remove the phase noise as in (3.13). As MLE relies on symbol decisions, its performance is degraded for larger amounts of residual phase noise, especially at low SNRs where many symbols decisions are incorrect.

Chapter 4

Digital Signal Processing Circuit Design

A coherent fiber-optic communication system can have a target symbol rate of 10 GBaud and higher. Typical DSPs used in coherent systems have a clock frequency in the order of 1 GHz, which means that the design will have to be extensively parallelized to meet throughput requirements. Furthermore, the DSP can account for a significant portion of the total system power dissipation [1], especially considering that processing previously performed in the analog domain is increasingly being replaced by DSP solutions. Thus, to keep the energy demand in check and reduce the need for cooling densely packed equipment, the power dissipation needs to be tightly controlled. Unfortunately, these two requirements are not always easy to combine, which makes the design of DSP circuits for fiber-optic systems challenging. In essence, the goal is to maximize the throughput using a limited power budget, while keeping the signal degradation that typically results from simplification of circuits as low as possible.

A general DSP system can be implemented using many different types of hardware, such as general-purpose processors (which might include specialized functions for DSP processing), ASICs or FPGAs. Due to the strict throughput requirements of fiber-optic communication systems, ASICs are the most common choice. However, FPGAs are often used as a development platform or for experimental systems.

Implementation of ASICs and FPGAs designs can be performed in different ways. The approach used in this work is low-level implementation using a hardware description language (HDL), in our case VHDL.

The HDL description of a design contains a formal representation of the functionality of the electronic circuit and can be used to simulate its bit-true behavior and to synthesize a netlist. The netlist contains information of the physical components of the ASIC or FPGA and how they are interconnected.

Besides HDL implementation, there are other methods that use higher abstraction levels, such as high-level synthesis (HLS). Here, the designer can work at the algorithmic level, using subsets of e.g. C or MATLAB, to generate register-transfer level HDL designs that can be used to synthesize a netlist. Using HLS can be more time efficient, as some circuit-level issues are abstracted away. However, direct HDL implementation gives the designer better control over circuit implementation details. In this chapter we describe the design process for ASICs and FPGAs and introduce selected important concepts related to circuit design and the methods used in this work.

4.1 Application-Specific Integrated Circuits

Application-specific integrated circuits (ASICs) are custom made for a particular purpose, and thus optimized to perform that task as efficiently as possible. The ASIC design and manufacturing process is complex, and the non-recurring engineering costs are high. However, when the design is finished and the manufacturing process is set up, the cost per unit can be relatively low. The performance, in terms of throughput for a coherent DSP, is higher and power dissipation is lower than for the FPGAs described below.

ASIC design of DSP systems typically use a cell-based very-large-scale integration (VLSI) design flow, where a cell library containing standard logic cells is included during synthesis. The cells describe logic functions, such as AND, OR and XOR, sequential functionality, such as flip-flops, or more complex functions such as full-adders or multiplexers. The library contains logic descriptions, data regarding timing and power dissipation, and the physical layout of each cell. When synthesizing a design, the synthesis tool transforms the HDL design into a netlist of interconnected standard cells, under constraints chosen by the designer. These constraints can be related to aspects such as timing, area or power dissipation. To manufacture an ASIC, the netlist needs to be taken through a place-and-route stage, where the cells in the netlist are placed onto the die and routed to create the electrical connections. A more detailed description of the cell-based design flow can be found in Paper D.

As an alternative to the cell-based design flow, a traditional full-custom VLSI flow can also be used. Here, the designer controls the layout and placement of each individual transistor in the circuit and manually draws the interconnects. This additional control and freedom can result in a more optimal circuit design, but it is very time consuming and thus most suitable for small or very regular designs. The two flow types can also be combined by using a full-custom approach only for the most critical parts of a design.

4.2 Field-Programmable Gate Arrays

Field-programmable gate arrays (FPGAs) are pre-fabricated, configurable circuits that can be used to implement a wide range of functionalities, including DSP. The main components enabling this functionality are the configurable logic blocks (CLBs) (Xilinx terminology) or adaptive logic modules (Altera) and routing resources with programmable interconnects [70]. The exact content of a CLB differs between FPGA models and vendors, but usually includes at least one multi-input lookup table, carry chains, registers and multiplexers. To increase the performance of operations that are hard to implement efficiently using the CLBs, specialized blocks are usually provided. These can include DSP slices, memories, specialized input/output controllers etc.

An FPGA is programmed rather than manufactured for each new design, and it can be reprogrammed and updated, resulting in much lower non-recurring engineering costs than for an ASIC. However, the flexibility comes at the cost of a lower maximum clock frequency and higher power dissipation than an ASIC implementing the same functionality, using the same process node. These factors limit the usability of FPGAs for high-speed coherent DSPs in deployed systems. Still, they are useful tools as evaluation and experimental platforms, and FPGA designs are often used as precursors for ASICs.

In theory, it is possible to use the same HDL design on both ASICs and FPGAs, providing that no specialized FPGA macros or ASIC IPs are used. In practice, the HDL code needs to be optimized separately for the different target systems. One reason is that writing HDL to best utilize the specialized functions onboard an FPGA, such as DSP blocks, might not translate well to an ASIC design. Another important aspect is the difference in clock rate, which will affect, e.g., the insertion of pipelining registers.

4.3 Circuit Design

This section is focused on digital complementary metal-oxide semiconductor (CMOS) ASIC design, but many of the topics are relevant also for FPGAs.

When designing a circuit there will always be trade-offs between timing, power dissipation and area (or resource utilization for FPGAs). Timing can often be considered the most important aspect here; in the case of DSP circuits, it determines if we reach our target throughput or not. The maximum clock rate at which we can run our system is determined by the critical path, which is the longest time delay between any elements in the design synchronized to a clock. By setting a timing constraint in synthesis, the tool will try to optimize the design, either using logic optimizations or by using larger library cells with a higher drive strength, until all paths are shorter than the constraint. If we fail to meet the timing constraint, a redesign of the HDL might be necessary.

Power dissipation for a CMOS ASIC can be divided into two major parts: static and dynamic. The former is related to leakage in the transistors and is mainly a problem for cell libraries using low-threshold voltages for performance reasons. Such libraries would most likely be used for DSP, which are expected to have high switching activity. For circuits that are expected to have low switching activity or for paths with large timing slacks, slower high-threshold voltage transistors can be used to reduce the static power dissipation [71]. The dynamic power dissipation is caused by the charging and discharging of the capacitance at the output of the logic gates as

$$P_{sw} = fV_{DD}^2 \sum_{i=1}^{N} C_i \alpha_i \tag{4.1}$$

where f is the clock rate, V_{DD} is the supply voltage, C_i and α_i are the capacitance and switching activity of the ith net, respectively. Netlist simulations, using appropriate test data, are used to capture the switching activity.

After synthesis, the capacitances of the cell pins are known and the capacitance of the interconnects can be estimated using a wire load model to perform power analysis [71]. Once place and route has been performed, the wire load can be extracted from the routed design to provide a better estimate of the dynamic power. In this thesis we estimate power dissipation after the synthesis stage, to avoid the many manual

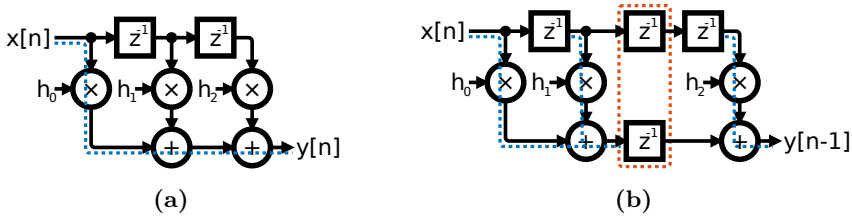

Figure 4.1: A three-tap FIR filter shown (a) without pipelining and with the critical path indicated by a blue dashed line, and (b) with one stage pipelining marked with an orange dashed square. Note that the pipelined version has three approximately equal shorter critical paths.

adjustments necessary for place and route, and the additional runtime. This approach means that the results can be sensitive to the accuracy of the wire load models. However, for the designs that have been taken through place and route, the agreement between post-place-and-route and post-synthesis power estimation has been good, which is expected for small designs [72].

4.3.1 Pipelining and Parallelism

As previously mentioned, the critical path of a design is what restricts the achievable clock rate, since the signals need to propagate through all logic elements on this path in one clock cycle. The path can be shortened by inserting additional pipelining stages, i.e. delay elements, at carefully chosen positions. By shortening the path, we can now increase the clock rate of the design at the cost of a larger latency, measured in clock cycles. However, care must be taken to retain the synchronization of different paths through the circuit. Fig. 4.1a shows an FIR filter, where the critical path is through one multiplier and two adders. By inserting a pipelining stage between the adders, as shown in Fig. 4.1b, the path can be shortened. However, an additional register also needs to be inserted in the top arm of the diagram to preserve the synchronization. Thus, the output is delayed by one clock cycle.

The addition of pipelining stages to a design has the additional advantage of reducing the probability of glitches, which are short unwanted signal toggles. Glitches can increase the dynamic power dissipation by increasing the switching activity α in (4.1) [73].

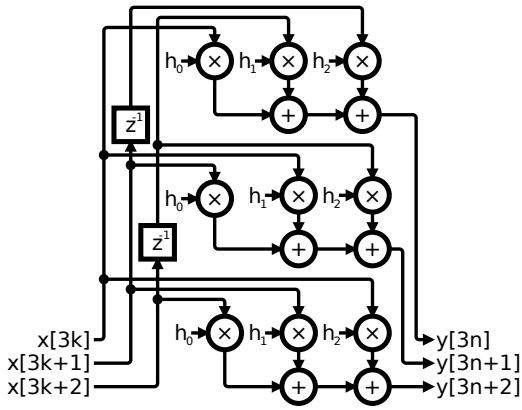

Figure 4.2: Three-tap FIR filter parallelized in three lanes.

For feed-forward circuits, pipelining is usually straightforward, as it does not affect the logic functionality of the circuit. For feedback circuits, the added cycle latency introduced by pipelining can affect the performance. An example of such a circuit is the dynamic equalizer described in Section 3.2.

Parallel processing is a way to reach a throughput higher than the clock rate by duplicating functional elements and processing multiple samples simultaneously. Fig. 4.2 shows a three-tap FIR filter parallelized in three lanes, resulting in three times higher throughput. Compared to the single lane filter in Fig. 4.1a, three times as many multipliers and adders are needed, illustrating a trade-off between area/power and throughput.

4.3.2 Clock Gating

In many circuit designs, parts of the circuit are not active all of the time. By applying clock gating, the clock signal to these parts is turned off when they are not in use. This saves power, as the clock tree can be a significant contributor to the total power dissipation. Additionally, the dynamic power dissipation (4.1) of the gated part will go to zero when no clock signal is present. The cost is an additional complexity of the clock net, which is often quickly offset by the power savings of introducing clock gating. The tap update of a pilot-based equalizer (Section 3.2.4) is a good example of where clock gating can be successfully employed. If there are enough samples between the pilots, the tap-update subcircuit can be clock gated when no pilot is received.

4.3.3 Resolution

A fixed-point number representation is often used for ASIC and FPGA implementations of DSP systems, as opposed to the floating-point representation commonly used for scientific computing. The main reason is the much smaller and less complex hardware designs necessary, resulting in a lower power dissipation and lower latency. The two most important properties of a fixed-point number are the wordlength, i.e. the number of bits, and the placement of the binary point. In a signed two's-complement representation of an all-integer 8-bit number,

$$\overbrace{0}^{\text{sign}}\ \overbrace{\underbrace{0}_{2^6}\ \underbrace{0}_{2^5}\ \underbrace{0}_{2^4}\ \underbrace{0}_{2^3}\ \underbrace{0}_{2^2}\ \underbrace{0}_{2^1}\ \underbrace{0}_{2^0}}^{\text{integer bits}}$$

the most significant bit (MSB) is reserved for sign representation and the signal can represent integer values between $-2^7 = -128$ to $2^7 - 2^0 = 127$. This can can also be referred to as an 8.0 representation.

A signed all-fraction 8-bit value, or 1.7 representation, still has a sign bit as the MSB, but the remaining bits are used to represent fractions, and can represent values between $-2^0 = -1$ to $2^0 - 2^{-7} = 0.992185$ in steps of $2^{-7} = 0.0078125$:

$$\overbrace{0}^{\text{sign}}\ .\ \overbrace{\underbrace{0}_{2^{-1}}\ \underbrace{0}_{2^{-2}}\ \underbrace{0}_{2^{-3}}\ \underbrace{0}_{2^{-4}}\ \underbrace{0}_{2^{-5}}\ \underbrace{0}_{2^{-6}}\ \underbrace{0}_{2^{-7}}}^{\text{fractional bits}}$$

The all-fraction representation has the useful property that multiplication of two such numbers results in a new all-fraction number, if we make sure that the -1 value is forbidden.

When designing a system utilizing a fixed-point number representation, the wordlength and placement of the binary point for each internal signal need to be carefully considered. To reduce area and power dissipation, the wordlengths should be as short as possible, but choosing the wordlength too small can result in degradation of the output quality. An additional challenge is the bit growth; when performing arithmetic operations on signals, the wordlengths might increase, depending on the type of operation. For multiplication

$$\begin{array}{rcc} & 0.10110 & 0.6875 \\ \times & 1.01111 & -0.53125 \\ \hline & 11.1010001010 & -0.365234375 \end{array}$$

the wordlength of the output is the sum of the wordlengths of the two inputs. If we assume that -1 is forbidden in the input values, the two MSBs both represent the sign, and one of them can be removed without losing precision. For addition and subtraction, we need to ascertain that we can store the full result in the output signal by adding an additional bit to the left, to avoid overflow:

$$\begin{array}{rrr} & 0.10110 & 0.6875 \\ - & 1.01111 & -0.53125 \\ \hline & 01.00111 & 1.21875 \end{array}$$

The bit growth needs to be handled during the design phase, either using truncation of the least significant bits or rounding. Truncation of a 6-bit signed fixed-point all-fraction number to four bits

$$\begin{array}{ll} 1.10111 & -0.28125 \\ 1.101 & -0.375 \end{array}$$

is similar to a *floor* operation. The result is rounding that is always downwards to the closest value that will fit in the new representation, which will introduce a negative bias to the rounded values. An alternative approach is to round the value before truncation by adding 1 to the MSB of the part that will be removed:

$$\begin{array}{rrr} & 1.10111 & -0.28125 \\ + & 0.0001 & 0.0625 \\ \hline & 1.110 & -0.25 \end{array}$$

This method has a small circuit implementation and works in most cases. However, a small positive bias is introduced as all half-way values are rounded up. For circuits where this bias can accumulate, methods such as convergent rounding are preferred. Here, the half-way values are rounded to the closest even (or odd) number, eliminating the bias at the cost of a slightly more complex circuit representation [30].

4.3.4 Mathematical Operations

Methods for circuit implementation of addition, subtraction and multiplication of fixed-point numbers are relatively straightforward, even though multiplication does not scale well to longer input wordlengths [74]. Division circuits are larger and slower than multiplication [75], and division

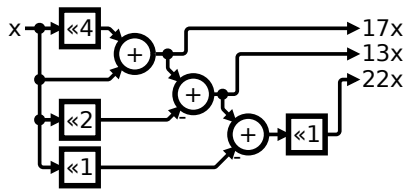

Figure 4.3: Example of MMCM.

operations are usually avoided in DSP designs if not strictly necessary. However, multiplication and division by powers of 2 are simple to implement as they correspond to bit-shift operations in a binary representation. A special case is multiplication with a constant, which can often be implemented in an efficient way using add/subtract and shift operations. For systems where one input value needs to be multiplied with a large number of constants, the benefit becomes even greater as the intermediate results can often be reused [76]. Fig. 4.3 shows an example of this method, called multiplierless multiple-constant multiplication (MMCM), where the input x is multiplied by three integer constants, 13, 17 and 22, by using only three adders. These types of circuit simplifications are often forgotten when estimating the complexity of DSP algorithms by counting the number of different operations needed.

For trigonometric functions, exponentials, logarithms and other functions, a direct circuit implementation can be very complex, if it exists. Instead, other methods need to be used, such as table lookup or approximations. A table lookup can be an efficient way to implement a mathematical function for shorter wordlengths, where the size of the table, stored in memory or realized using logic functions, can be limited. In cases where the table becomes too large, the function can instead be approximated e.g. by using a truncated Taylor series. The accuracy of these approximations depends on the degree of the polynomial used, and can quickly become more complex than a table lookup if a high-degree polynomial is necessary.

Trigonometric functions can be calculated using a coordinate rotation digital computer (CORDIC) circuit [77]. This iterative circuit can be used to calculate e.g. sin, cos, arctan and the magnitude of a complex value. For each iteration i, the operations

$$\begin{aligned} x_{i+1} &= x_i - \mu d_i y_i 2^{-i} \\ y_{i+1} &= y_i + d_i x_i 2^{-1} \\ z_{i+1} &= z_i - d_i \alpha_i, \end{aligned} \tag{4.2}$$

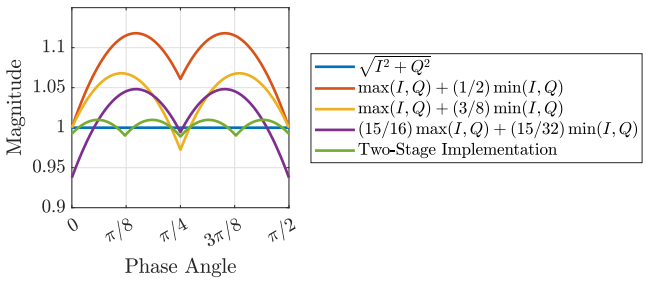

Figure 4.4: Magnitude calculated using different parameter settings for the $\alpha \max + \beta \min$ approximator.

are performed, where x_0, y_0 and z_0 are the inputs and parameters μ, d and α are selected based on the type of operation. To calculate the magnitude and phase of a complex value $A = I + jQ$, the following parameters can be used:

$$\begin{aligned} x_0 &= I & \mu &= 1 \\ y_0 &= Q & d_i &= -\operatorname{sgn}(x_i y_i) \\ z_0 &= 0 & \alpha_i &= \arctan(2^{-i}) \end{aligned}$$

Given enough iterations, the circuit will converge to $x_k = K|A|$, $y_k = 0$ and $z_k = \arctan(Q/I)$, where K is a constant that depends on the number of iterations performed. As seen in (4.2), an efficient CORDIC circuit can be realized using only additions and bit-shift operations. However, loop-unrolling is often necessary in DSP implementations to reach the target throughput.

To calculate the magnitude of a complex number, an alternative approach is the $\alpha \max + \beta \min$ approximation [30]. This method is used to avoid the square root in the expression $|A| = \sqrt{I^2 + Q^2}$, which can result in a complex circuit. Compared to CORDIC, the $\alpha \max + \beta \min$ circuit is smaller and faster, but with a less precise result. The magnitude is approximated as

$$|A| \approx \alpha \max(I, Q) + \beta \min(I, Q), \tag{4.3}$$

where α and β are selected so that the circuit can be implemented using additions and bit-shift operations only. The approximation error is dependent on the phase angle of A, as shown in Fig. 4.4 for a selection of possible implementations. The approximation can be improved by using

a two-stage approach,

$$\begin{aligned} m_0 &= \frac{127}{128}\max(I,Q) + \frac{3}{16}\min(I,Q), \\ m_1 &= \frac{27}{32}\max(I,Q) + \frac{71}{128}\min(I,Q), \\ |A| &\approx \max(m_0, m_1), \end{aligned} \tag{4.4}$$

at the cost of a larger circuit.

4.4 Circuit Verification and Evaluation

The performance of a circuit in terms of output quality can be measured in many ways, and the metric depends on the circuit under test. For the circuits presented in this thesis, the bit-error rate (BER) and the error-vector magnitude (EVM) are mostly used. The BER is simply the number of bit errors in the received data divided by the total number of transmitted bits, resulting in a probability that a received bit is incorrectly decoded. A related term is SNR penalty, which is how much higher SNR the evaluated system needs in order to reach the same BER as some reference, usually an AWGN channel. The EVM can be calculated as the root mean square (RMS) error between the received symbols x and the ideal symbols r, usually normalized to the RMS of the ideal symbols

$$EVM_{RMS} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}|x_i - r_i|^2}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}|r_i|^2}}, \tag{4.5}$$

where N is the number of symbols considered.

To record these metrics for an HDL description of a DSP subsystem, software-based logic simulation of the HDL is often used, as it captures all aspects of the logic functionality of the circuit, including effects of limited wordlengths and any approximations. Input data to these simulations needs to be generated using a suitable channel model and the output can be exported from the HDL simulation for analysis in a fitting tool. In this thesis we employ MATLAB-HDL co-simulation, where the test data is generated using floating-point representation in MATLAB, converted to fixed-point and exported to the HDL simulator. After simulation, the HDL output is imported back into MATLAB for analysis and calculation of e.g. BER or EVM.

4.4.1 Fiber-on-Chip

The software-based simulation method works well for evaluation of metrics that do not require a large number of samples to reach reliable results. For designs that have long convergence times, for very low BER simulations, when evaluating phenomena that are very rare (e.g. cycle slips, Sec. 3.3.1) or when studying long-term stability, the runtimes can be prohibitively long. As an alternative to software-based simulations, we have developed the Fiber-on-Chip (FoC) approach, where we emulate a digital communication system, including data generation and channel models, onboard an FPGA or ASIC. The resulting speed-up is multiple orders of magnitude compared to software-based simulations. FoC can not replace real fiber experiments with real-time DSP. However, it is quicker to set up and it offers some interesting features, such as repeatability. By controlling the seeds of the random generators used for data and impairment generation, different solutions can be evaluated long-term using identical input data.

Our HDL implementation of the FoC approach, CHOICE, supports multiple modulation formats from QPSK to 256QAM and emulation of dual-polarization systems. It includes channel emulators for AWGN, phase noise, PMD and SOP rotations, with the possibility of varying any parameter with time. Additionally, events can be introduced to emulate e.g. fast SOP changes for a limited number of samples. The transmitter part include modules for data generation, modulation and pulse shaping, and is used to generate synthetic data as input to the fiber model. On the receiver side, a range of analysis tools are available, such as error counters and cycle-slip detectors. The CHOICE system is modular and easy to extend with additional functionality. Examples of extensions include a Kerr-effect emulator and a waveform memory, described in Papers H and N, respectively. The memory can be used to replay data captured in physical experiments or to capture long data series from any signal of the system. The FoC approach and CHOICE are described in detail in Paper VI. CHOICE has been release under an open source license.

Chapter 5

Summary of Contributions

This chapter will describe the main contributions presented in the publications included in this thesis. The first four publications are related to our work on circuit implementation of CPR subsystems for coherent fiber-optical communication systems, highlighting the trade-offs related to converting an algorithm description to an actual circuit. Paper V describes a MIMO equalizer aimed at systems employing CC-MCF. The final paper (Paper VI) describes the FoC approach to circuit evaluation.

In **Paper I**, which is an extension of the work done in Paper B, we present an implementation of the BPS CPR algorithm and describe the optimizations and modifications performed on the original algorithm to reach a working VLSI circuit design. We show how the complexity of the otherwise very demanding phase rotations can be reduced by mapping all symbols to the first quadrant and utilizing MMCM for the rotation operations. Furthermore, we show how a very small interpolation circuit can be used to reduce the number of test phases, and how block averaging can replace the sliding window to simplify parallel implementation. The paper also includes an evaluation of how design parameters affect the SNR penalty at the output. For 16QAM we reach an energy efficiency of 1 pJ/bit, using a 22 nm CMOS library and a supply voltage of 0.8 V. However, scaling to higher-order formats results in higher energy per bit due to the larger number of test angles and higher wordlengths needed. As a comparison, we present a simple pilot-based CPR with a better energy efficiency. However, this improvement comes at the cost of a higher SNR penalty and lower spectral efficiency.

Paper II describes CPR circuit implementations for higher-order QAM formats, focusing on 256QAM. We include both one- and two-stage

solutions utilizing modified Viterbi–Viterbi estimators, PCPE and BPS. We describe the optimizations performed on the circuits and how design parameters can be selected to maximize performance. The CPR circuits are evaluated in terms of energy efficiency and SNR penalty, where the two-stage solutions generally results in a better trade-off between the two. Utilizing a modified Viterbi–Viterbi estimator or PCPE as a first stage, with a simplified BPS as a second stage makes it possible to reach 1 pJ/bit with an SNR penalty of 0.6 dB, using a 22 nm library at 0.8 V. However, the PCPE+BPS version is preferred due to its smaller area. Versions of the PCPE circuit described in this paper have been used in the experiments presented in Papers K–M and others.

To identify the optimal choice of CPR implementation for the most common QAM formats, **Paper III** presents benchmarks for multiple one and two-stage circuit implementations. The circuits are evaluated for 16QAM, 64QAM and 256QAM in terms of SNR penalty, energy efficiency and linewidth sensitivity. We also evaluate the cycle-slip probability using our FoC system, where the PCPE shows a very low cycle-slip probability for $\Delta v T_S < 10^{-5}$. Our benchmarks show that single-stage methods are preferred for 16QAM, while the higher power dissipation of the two-stage methods can be motivated for 64QAM and 256QAM due to the lower SNR penalty. We also include calculations of the latencies of the circuits, should they be used in the feedback loop of a decision-directed equalizer. Compared to the single-stage circuits, the latency of the two-stage implementations are much higher due to the longer pipeline and the two averaging operations performed.

In **Paper IV**, we describe a multi-format CPR circuit based on PCPE, that in one single circuit can support QPSK, 16QAM, 32QAM and 64QAM. To reduce the effective wordlength of internal signals when lower-order formats are used, clock-gating and signal masks are used extensively throughout the circuit. By eliminating switching in this way, the power dissipation for QPSK processing can be kept much lower than for 64QAM, even though the same circuit is used. Calculation of the principal component for 32QAM is less exact as the format does not contain any outer corner points, which can result in poor performance. The addition of a magnitude scaling of the innermost ring in the constellation, which is implemented using a low-complexity circuit, reduces this problem. An added benefit is improved performance also for the other modulation formats.

Paper V introduces a pilot-based MIMO equalizer circuit for coupled-core multi-core fiber transmission. The circuit is an LMS equalizer using pilot symbols with variable insertion ratio. The design is parameterized to support the generation of $N \times N$ MIMO filters with adjustable parallelization. We use this circuit to study convergence behavior in conjunction with a temporally drifting channel model, and to evaluate the feasibility of MIMO equalization for higher-dimensionality SDM systems, using cell area as a metric. This implementation could be used as a benchmark for further DSP development and optimization, as it indicates where development effort is best spent.

Extending the work done in Papers C and G, **Paper VI** provides a description of the Fiber-on-Chip (FoC) approach to circuit verification, including a specification of our HDL implementation of the approach. This system allows us to model a complete fiber-optic system onboard an FPGA or ASIC, including AWGN, phase noise and PMD effects. The paper includes descriptions and evaluations of the channel models. As a demonstration of the capabilites, two case studies are included where we evaluate a CMA equalizer. In the first demonstration we run the system on an FPGA to study how the equalizer responds to sudden PMD events. In the second study, the FoC system is used to generate input data to a CMA equalizer onboard a manufactured ASIC. Here, the CMA is implemented in a separate power domain, which makes it possible to produce credible power measurements without the need to interface with an optical setup.

References

[1] B. S. G. Pillai, B. Sedighi, K. Guan, N. P. Anthapadmanabhan, W. Shieh, K. J. Hinton, and R. S. Tucker, "End-to-end energy modeling and analysis of long-haul coherent transmission systems," *Journal of Lightwave Technology*, vol. 32, no. 18, pp. 3093–3111, June 2014.

[2] R. Wood and J. Konieczny, "Fixed network data traffic: Worldwide trends and forecasts 2019–2025," Analysis Mason, Tech. Rep., 2020.

[3] E. Agrell, M. Karlsson, A. R. Chraplyvy, D. J. Richardson, P. M. Krummrich, P. Winzer, K. Roberts, J. K. Fischer, S. J. Savory, B. J. Eggleton, M. Secondini, F. R. Kschischang, A. Lord, J. Prat, I. Tomkos, J. E. Bowers, S. Srinivasan, M. Brandt-Pearce, and N. Gisin, "Roadmap of optical communications," *Journal of Optics*, vol. 18, no. 6, p. 063002, May 2016.

[4] T. H. Maiman, "Stimulated optical radiation in ruby," *Nature*, vol. 187, no. 4736, pp. 493–494, Aug. 1960.

[5] K. C. Kao and G. A. Hockham, "Dielectric-fibre surface waveguides for optical frequencies," *Proceedings of the Institution of Electrical Engineers*, vol. 113, no. 7, pp. 1151–1158, June 1966.

[6] F. P. Kapron, D. B. Keck, and R. D. Maurer, "Radidation losses in glass optical waveguides," *Applied Physics Letters*, vol. 17, no. 10, p. 423, Nov. 1970.

[7] R. J. Mears, L. Reekie, I. M. Jauncey, and D. N. Payne, "Low-noise erbium-doped fibre amplifier operating at 1.54 µm," *Electronics Letters*, vol. 23, no. 19, pp. 1026–1028, Sept. 1987.

[8] M. G. Taylor, "Coherent detection method using DSP for demodulation of signal and subsequent equalization of propagation impairments," *IEEE Photonics Technology Letters*, vol. 16, no. 2, pp. 674–676, Feb. 2004.

[9] E. Maniloff, S. Gareau, and M. Moyer, "400G and beyond: Coherent evolution to high-capacity inter data center links," in *Optical Fiber Communication Conference (OFC)*, Mar. 2019.

[10] B. J. Puttnam, G. Rademacher, and R. S. Luís, "Space-division multiplexing for optical fiber communications," *Optica*, vol. 8, no. 9, pp. 1186–1203, Sep 2021.

[11] R. S. Williams, "What's next? [The end of Moore's law]," *Computing in Science & Engineering*, vol. 19, no. 2, pp. 7–13, Mar. 2017.

[12] R. Ryf, A. Sierra, R.-J. Essiambre, A. H. Gnauck, S. Randel, M. Esmaeelpour, S. Mumtaz, P. J. Winzer, R. Delbue, P. Pupalaikis, A. Sureka, T. Hayashi, T. Taru, and T. Sasaki, "Coherent 1200-km 6 × 6 MIMO mode-multiplexed transmission over 3-core microstructured fiber," in *European Conference on Optical Communications (ECOC)*, Sept. 2011, p. Th.13.C.1.

[13] G. P. Agrawal, *Nonlinear Fiber Optics*, 5th ed. Academic Press, 2013.

[14] M. Safak, *Digital Communications*, 1st ed. Newark: John Wiley & Sons, 2017.

[15] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, July 1948.

[16] P. Krummrich, E.-D. Schmidt, W. Weiershausen, and A. Mattheus, "Field trial results on statistics of fast polarization changes in long haul WDM transmission systems," in *Optical Fiber Communication Conference (OFC)*, vol. 4, Mar. 2005.

[17] J. P. Gordon and H. Kogelnik, "PMD fundamentals: Polarization mode dispersion in optical fibers," *Proceedings of the National Academy of Sciences*, vol. 97, no. 9, pp. 4541–4550, Apr. 2000.

[18] C. B. Czegledi, M. Karlsson, P. Johannisson, and E. Agrell, "Temporal stochastic channel model for absolute polarization state and polarization-mode dispersion," in *Optical Fiber Communications Conference (OFC)*, Mar. 2017.

[19] G. P. Agrawal, *Fiber-Optic Communication Systems*, 4th ed. Wiley, 2010.

[20] G. Foschini and G. Vannucci, "Characterizing filtered light waves corrupted by phase noise," *IEEE Transactions on Information Theory*, vol. 34, no. 6, pp. 1437–1448, Nov. 1988.

[21] W. Shieh and K.-P. Ho, "Equalization-enhanced phase noise for coherent-detection systems using electronic digital signal processing," *Optics Express*, vol. 16, no. 20, pp. 15 718–15 727, Sep 2008.

[22] A. Arnould and A. Ghazisaeidi, "Equalization enhanced phase noise in coherent receivers: DSP-aware analysis and shaped constellations," *Journal of Lightwave Technology*, vol. 37, no. 20, pp. 5282–5290, Oct. 2019.

[23] A. Kakkar, J. R. Navarro, R. Schatz, X. Pang, O. Ozolins, H. Louchet, G. Jacobsen, and S. Popov, "Mitigation of EEPN in coherent optical systems with low-speed digital coherence enhancement," *IEEE Photonics Technology Letters*, vol. 27, no. 18, pp. 1942–1945, Sept. 2015.

[24] M. Ohtsu and S. Kotajima, "Linewidth reduction of a semiconductor laser by electrical feedback," *IEEE Journal of Quantum Electronics*, vol. 21, no. 12, pp. 1905–1912, Dec. 1985.

[25] M. S. Faruk and S. J. Savory, "Digital signal processing for coherent transceivers employing multilevel formats," *Journal of Lightwave Technology*, vol. 35, no. 5, pp. 1125–1141, 2017.

[26] F. Gardner, "A BPSK/QPSK timing-error detector for sampled receivers," *IEEE Transactions on Communications*, vol. 34, no. 5, pp. 423–429, May 1986.

[27] K. Mueller and M. Muller, "Timing recovery in digital synchronous data receivers," *IEEE Transactions on Communications*, vol. 24, no. 5, pp. 516–531, May 1976.

[28] M. Morelli and U. Mengali, "Feedforward frequency estimation for PSK: A tutorial review," *European Transactions on Telecommunications*, vol. 9, no. 2, pp. 103–116, Sept. 1998.

[29] A. Leven, N. Kaneda, U.-V. Koc, and Y.-K. Chen, "Frequency estimation in intradyne reception," *IEEE Photonics Technology Letters*, vol. 19, no. 6, pp. 366–368, Mar. 2007.

[30] R. G. Lyons, *Understanding Digital Signal Processing*, 3rd ed. Prentice Hall, 2011.

[31] S. J. Savory, "Digital filters for coherent optical receivers," *Optics Express*, vol. 16, no. 2, pp. 804–817, Jan 2008.

[32] J. Wang, C. Xie, and Z. Pan, "Matched filter design for RRC spectrally shaped Nyquist-WDM systems," *IEEE Photonics Technology Letters*, vol. 25, no. 23, pp. 2263–2266, Dec. 2013.

[33] M. Paskov, D. Lavery, and S. J. Savory, "Blind equalization of receiver in-phase/quadrature skew in the presence of Nyquist filtering," *IEEE Photonics Technology Letters*, vol. 25, no. 24, pp. 2446–2449, Dec. 2013.

[34] S. S. Haykin, *Adaptive filter theory*, ser. Prentice-Hall information and system sciences series. Prentice Hall, 1996.

[35] T. Sakamoto, T. Mori, M. Wada, T. Yamamoto, F. Yamamoto, and K. Nakajima, "Strongly-coupled multi-core fiber and its optical characteristics for MIMO transmission systems," *Optical Fiber Technology*, vol. 35, pp. 8–18, Feb. 2017.

[36] D. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1867–1875, Nov. 1980.

[37] W. Sethares, G. Rey, and C. Johnson, "Approaches to blind equalization of signals with multiple modulus," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, May 1989, pp. 972–975.

[38] I. Fatadin, D. Ives, and S. J. Savory, "Blind equalization and carrier phase recovery in a 16-QAM optical coherent system," *Journal of Lightwave Technology*, vol. 27, no. 15, pp. 3042–3049, Aug 2009.

[39] M. S. Faruk, Y. Mori, C. Zhang, and K. Kikuchi, "Proper polarization demultiplexing in coherent optical receiver using constant modulus algorithm with training mode," in *OECC 2010 Technical Digest*, July 2010, pp. 768–769.

[40] L. Liu, Z. Tao, W. Yan, S. Oda, T. Hoshida, and J. C. Rasmussen, "Initial tap setup of constant modulus algorithm for polarization demultiplexing in optical coherent receivers," in *Conference on Optical Fiber Communication*, Mar. 2009.

[41] M. Ready and R. Gooch, "Blind equalization based on radius directed adaptation," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, Apr. 1990, pp. 1699–1702.

[42] D. Lavery, M. Paskov, R. Maher, S. J. Savory, and P. Bayvel, "Modified radius directed equaliser for high order QAM," in *European Conference on Optical Communication (ECOC)*, Sept. 2015, pp. 1–3.

[43] J. G. Proakis, *Digital Communications*, 4th ed. Boston: McGraw Hill, 2001.

[44] T. Pfau, "Real-time implementation of high-speed digital coherent transceivers," in *Enabling Technologies for High Spectral-efficiency Coherent Optical Communication Networks*, X. Zhou and C. Xie, Eds. Wiley Telecom, 2016, ch. 12, pp. 435–446.

[45] M. Mazur, J. Schröder, A. Lorences-Riesgo, T. Yoshida, M. Karlsson, and P. A. Andrekson, "Overhead-optimization of pilot-based digital signal processing for flexible high spectral efficiency transmission," *Opt. Express*, vol. 27, no. 17, pp. 24 654–24 669, Aug 2019.

[46] X. Zhou and J. Yu, "Multi-level, multi-dimensional coding for high-speed and high-spectral-efficiency optical transmission," *Journal of Lightwave Technology*, vol. 27, no. 16, pp. 3641–3653, Aug. 2009.

[47] J. Li, L. Li, Z. Tao, T. Hoshida, and J. C. Rasmussen, "Laser-linewidth-tolerant feed-forward carrier phase estimator with reduced complexity for QAM," *Journal of Lightwave Technology*, vol. 29, no. 16, pp. 2358–2364, Aug. 2011.

[48] X. Zhou and J. Yu, "Two-stage feed-forward carrier phase recovery algorithm for high-order coherent modulation formats," in *European Conference on Optical Communication (ECOC)*, Sept. 2010.

[49] T. Pfau and R. Noé, "Phase-noise-tolerant two-stage carrier recovery concept for higher order QAM formats," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 16, no. 5, pp. 1210–1216, Dec 2010.

[50] S. M. Bilal, C. R. S. Fludger, V. Curri, and G. Bosco, "Multistage carrier phase estimation algorithms for phase noise mitigation in 64-quadrature amplitude modulation optical systems," *Journal of Lightwave Technology*, vol. 32, no. 17, pp. 2973–2980, Sep. 2014.

[51] C. S. Martins, F. P. Guiomar, and A. N. Pinto, "Hardware optimization of dual-stage carrier-phase recovery for coherent optical receivers," *OSA Continuum*, vol. 4, no. 12, pp. 3157–3175, Dec 2021.

[52] E. Ip and J. M. Kahn, "Feedforward carrier recovery for coherent optical communications," *Journal of Lightwave Technology*, vol. 25, no. 9, pp. 2675–2692, Sept. 2007.

[53] C. R. S. Fludger, D. Nuss, and T. Kupfer, "Cycle-slips in 100G DP-QPSK transmission systems," in *Optical Fiber Communication Conference (OFC)*, Mar. 2012.

[54] M. G. Taylor, "Phase estimation methods for optical coherent detection using digital signal processing," *Journal of Lightwave Technology*, vol. 27, no. 7, pp. 901–914, Apr. 2009.

[55] H. Cheng, Y. Li, F. Zhang, J. Wu, J. Lu, G. Zhang, J. Xu, and J. Lin, "Pilot-symbols-aided cycle slip mitigation for DP-16QAM optical communication systems," *Opt. Express*, vol. 21, no. 19, pp. 22 166–22 172, Sep 2013.

[56] Y. Gao, E. Ha, A. P. T. Lau, C. Lu, X. Xu, and L. Li, "Non-data-aided and universal cycle slip detection and correction for coherent communication systems," *Opt. Express*, vol. 22, no. 25, pp. 31 167–31 179, Dec 2014.

[57] H. Meyr, M. Moeneclaey, and S. Fechtel, *Digital Communication Receivers, Volume 2: Synchronization, Channel Estimation, and Signal Processing*, ser. Wiley Series in Telecommunications and Signal Processing. Wiley, 1997.

[58] A. Spalvieri and L. Barletta, "Pilot-aided carrier recovery in the presence of phase noise," *IEEE Transactions on Communications*, vol. 59, no. 7, pp. 1966–1974, July 2011.

[59] M. Magarini, L. Barletta, A. Spalvieri, F. Vacondio, T. Pfau, M. Pepe, M. Bertolini, and G. Gavioli, "Pilot-symbols-aided carrier-phase recovery for 100-G PM-QPSK digital coherent receivers," *IEEE Photonics Technology Letters*, vol. 24, no. 9, pp. 739–741, May 2012.

[60] A. Viterbi and A. Viterbi, "Nonlinear estimation of PSK-modulated carrier phase with application to burst digital transmission," *IEEE Transactions on Information Theory*, vol. 29, no. 4, pp. 543–551, July 1983.

[61] M. Seimetz, "Laser linewidth limitations for optical systems with high-order modulation employing feed forward digital carrier phase estimation," in *Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference*, Feb. 2008.

[62] Y. Gao, A. P. T. Lau, C. Lu, J. Wu, Y. Li, K. Xu, W. Li, and J. Lin, "Low-complexity two-stage carrier phase estimation for 16-QAM systems using QPSK partitioning and maximum likelihood detection," in *Optical Fiber Communication Conference (OFC)*, Mar. 2011.

[63] I. Fatadin, D. Ives, and S. J. Savory, "Laser linewidth tolerance for 16-QAM coherent optical systems using QPSK partitioning," *IEEE Photonics Technology Letters*, vol. 22, no. 9, pp. 631–633, May 2010.

[64] J. H. Ke, K. P. Zhong, Y. Gao, J. C. Cartledge, A. S. Karar, and M. A. Rezania, "Linewidth-tolerant and low-complexity two-stage carrier phase estimation for dual-polarization 16-QAM coherent optical fiber communications," *Journal of Lightwave Technology*, vol. 30, no. 24, pp. 3987–3992, Dec. 2012.

[65] S. M. Bilal, G. Bosco, P. Poggiolini, and C. R. S. Fludger, "Low-complexity linewidth-tolerant carrier phase estimation for 64-QAM systems based on constellation transformation," in *European Conference on Optical Communication (ECOC)*, Sept. 2013, pp. 1–3.

[66] T. Pfau, S. Hoffmann, and R. Noe, "Hardware-efficient coherent digital receiver concept with feedforward carrier recovery for M-QAM constellations," *Journal of Lightwave Technology*, vol. 27, no. 8, pp. 989–999, Apr. 2009.

[67] H. Sun, K. Wu, S. Thomson, and Y. Wu, "Novel 16QAM carrier recovery based on blind phase search," in *European Conference on Optical Communication (ECOC)*, Sep. 2014.

[68] J. C. M. Diniz, Q. Fan, S. M. aes Ranzini, F. N. Khan, F. D. Ros, D. Zibar, and A. P. T. Lau, "Low-complexity carrier phase recovery based on principal component analysis for square-QAM modulation formats," *Optics Express*, vol. 27, no. 11, pp. 15 617–15 626, May 2019.

[69] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, July 2010.

[70] M. A. Frefzer and A. M. Tyrrell, *Evolvable Hardware: From Practice to Application*, ser. Natural Computing Series. New York: Springer, 2015.

[71] S. P. Khatri, N. V. Shenoy, J.-C. Giomi, and A. Khouja, "Logic synthesis," in *Electronic Design Automation for IC Implementation, Circuit Design, and Process Technology*, L. Lavagno, G. Martin, I. L. Markov, and L. K. Scheffer, Eds. Taylor & Francis, 2016, ch. 1, pp. 4–24.

[72] K. D. Boese, A. B. Kahng, and S. Mantik, "On the relevance of wire load models," in *Proceedings of the 2001 International Workshop on System-Level Interconnect Prediction*, ser. SLIP 01. New York, NY, USA: Association for Computing Machinery, Mar. 2001, pp. 91–98.

[73] C. Tsui, M. Pedram, and A. M. Despain, "Efficient estimation of dynamic power consumption under a real delay model," in *Proceedings of 1993 International Conference on Computer Aided Design (ICCAD)*, Nov. 1993, pp. 224–228.

[74] M.-B. Lin, *Introduction to VLSI Systems: A Logic, Circuit, and System Perspective*. CRC Press, 2012.

[75] M. D. Ercegovac and T. Lang, *Digital Arithmetic*. Morgan Kaufmann Publishers, 2004.

[76] Y. Voronenko and M. Püschel, "Multiplierless multiple constant multiplication," *ACM Transactions on Algorithms*, vol. 3, no. 2, p. 11, May 2007.

[77] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Transactions on Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, Sept. 1959.

Papers I–VI