



## **Challenges and Trade-offs in Real-Time Implementation of DSP for Coherent Transmission**

Downloaded from: <https://research.chalmers.se>, 2025-12-04 13:38 UTC

Citation for the original published paper (version of record):

Larsson-Edefors, P. (2020). Challenges and Trade-offs in Real-Time Implementation of DSP for Coherent Transmission. Optics InfoBase Conference Papers, Part F191-SPPCom 2020

N.B. When citing this work, cite the original published paper.

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

# Challenges and Trade-offs in Real-Time Implementation of DSP for Coherent Transmission

Per Larsson-Edefors

*Dept. of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden  
perla@chalmers.se*

**Abstract:** We review different real-time implementation platforms for digital signal processing. We discuss circuit implementation of coherent receivers and design trade-offs involving circuit complexity, throughput and power dissipation. © 2020 The Author(s)

**OCIS codes:** (060.0060) Fiber optics and optical communication;(060.1660) Coherent communications

## 1. Introduction

Real-time implementation can give designers insights in how DSP functions behave in realistic systems, in which channel properties vary over time and infrequent, bursty events occasionally take place. For optical communication systems, evaluating deep bit-error rates (BERs) is a methodological problem to which real-time DSP and its ability to accelerate BER analysis is a remedy. An additional benefit of real-time DSP is that as hardware descriptions of algorithms are being developed, digital architectures selected, and fixed-point number resolutions established, we obtain sufficient information to estimate accurately power and energy dissipation of DSP functions.

## 2. Real-Time Implementation Platforms

Two different platforms are available for real-time implementations: Application-specific integrated circuits (ASICs) and field-programmable gate arrays (FPGAs). Advantages of FPGA prototyping include relatively short development times [1] and, possibly, integration of analog-to-digital converters (ADCs) [2]. However, because of their limited logic resources, FPGAs can accommodate only smaller systems, e.g. only one polarization out of two [3]. In contrast, ASICs may appear to have unlimited resources. For example, in 2020, NVIDIA announced the Ampere 7-nm GA100 GPU with 54 billion transistors on a 826-mm<sup>2</sup> die [4]. But large chips are clearly very expensive to develop and factors like chip fabrication yield eventually limit the die size of monolithic chips.

Once bipolar technologies were necessary to reach sufficient ADC bandwidths [5] and it was natural to combine technology-diverse ASICs and commercial-off-the-shelf chips and components. But the limited electrical chip-to-chip bandwidth is a serious impediment to system integration of complex real-time implementations. Thanks to time-interleaved successive-approximation (SAR) ADC architectures and their ability to scale with (digital) CMOS technology, CMOS ADCs can reach high sampling rates [6] enabling on-chip DSP-ADC integration [7]. While on-chip integration greatly enhances the signal bandwidth between different units, the noise injected by high-speed switching logic circuits into sensitive analog portions, like ADC samplers, presents a design challenge.

## 3. DSP for Intradyne Coherent Receivers

Since it has to adapt to varying channel properties, the receiver DSP is considerably more complex to implement than the transmitter DSP. Adaptivity challenges implementation in two ways: 1) Logic gate delays limit the data throughput of circuits with feedback. Unless increased loop cycle latencies can be accepted by the DSP algorithm, delay minimization is required, but this rapidly increases power dissipation. 2) Adaptivity consumes resources. For example, adaptive FIR filters require full  $n$ -bit  $\times$   $m$ -bit tap multipliers. But if the  $m$ -bit filter coefficients are static, the tap multipliers can be significantly simplified, reducing logic gate usage, delay and power dissipation.

In a polarization- and phase-diverse intradyne receiver, four ADCs sample the incoming data; i.e., I and Q channels of two polarizations. Static chromatic dispersion compensation (CDC) is performed on each polarization, after which an adaptive equalizer (AE) handles residual chromatic dispersion and performs polarization demultiplexing. The carrier phase recovery (CPR) removes phase noise and, finally, forward-error correction (FEC) decoding reduces the receiver output BER down to  $10^{-15}$ . Apart from DSP and FEC, functions to compensate ADC bandwidth limitations, correct skew between incoming channels, align the sampling clock (timing recovery), etc., are required. Also supporting digital circuits, such as buffers and interleavers, are required. While not that interesting from an algorithmic perspective, such circuits are resource demanding and can cause bandwidth problems.

Going backwards through the receiver, we find the complex FEC decoder which stores data blocks in a large memory. In contrast to DSP, which performs operations on *all* modulated symbols, the main function of FEC is to monitor the demodulated data stream. The FEC decoder performs an operation on the data when an error is detected, but this is a *relatively rare event*. Thanks to this principle of operation, we can trade circuit area for lower power dissipation in FEC decoders [8, 9]. Interestingly, this trade-off is not available in DSP implementations.

In the DSP chain, the relative complexities of CPR, AE, and CDC units vary with fiber reach. It was shown for a 16QAM datacenter interconnect that the AE unit is dominating [10]. For shorter reaches, for which we can neglect

chromatic and polarization-mode dispersion, CDC is not needed, making CPR relatively more important. But regardless of reach, the AE unit is complex to implement, because of its adaptive taps which continuously adjust to compensate different linear impairments. By reconsidering how error and tap update calculations are performed, the AE unit implementation can be simplified [11]: For example, in shorter fibers, since the polarization rotation slows down [12], the AE tracking speed can be reduced, relaxing the requirement on the tap update feedback.

It is challenging to implement ADCs with high sampling rates. While digital subcarrier multiplexing offers one path to limiting symbol rate at the system level [3, 13], choice of oversampling is a trade-off available during receiver implementation. An oversampling of 2 samples per symbol (SPS) has been common, since this choice relaxes ADC requirements and enables powerful equalization schemes [7]. If the oversampling rate is reduced, penalties due to aliasing and reduced filter bandwidths increase. Approaching 1 SPS would be good from an ADC and AE power dissipation perspective, but this puts impractically strict requirements on the sampling time jitter.

#### 4. Real-Time Prototyping of Subsystems

Real-time subsystem prototypes can give design insights not possible to obtain from time-consuming simulations. But testing strategies for DSP and FEC are very different: As FEC uses demodulated binary data, on-chip random-data generators can be used to test real-time FEC prototypes, even those using soft information from ADCs [8, 14]. Regardless of platform, FPGA [8] or ASIC [14], it is essential to keep all high-speed signaling internal to the prototype. External low-speed signals, however, can be used to configure the on-chip data generators.

Because DSP operates on modulated signals, it is challenging to develop testing methodologies for real-time DSP prototypes. Off-chip signal sources lead to bandwidth problems, while on-chip memories, which can store realistic waveforms, are limited in capacity. Recently an approach that can digitally emulate a fiber system including channel impairments [15] was used to perform cycle-slip evaluations of CPR circuits [16].

#### 5. Conclusion

We have reviewed two implementation platforms for real-time DSP, viz. ASICs and FPGAs, and discussed trade-offs associated with implementation of coherent receivers. Chip-to-chip bandwidth limitations make implementation of real-time prototypes difficult. In this respect, advanced FPGAs with integrated ADCs simplify implementation, however, FPGAs are limited in logic resources. ASIC prototyping offers flexibility and performance, but is very costly. Digitally emulated channel impairments offer a cost-effective path to real-time testing of subsystems.

#### References

1. C. R. S. Fludger, J. C. Geyer, T. Duthel, S. Wiese, and C. Schulien, "Real-time prototypes for digital coherent receivers," in *Opt. Fiber Commun. Conf.*, (2010), p. OMS1.
2. B. Farley *et al.*, "A programmable RFSoc in 16nm FinFET technology for wideband communications," in *IEEE Asian Solid-State Circuits Conf.*, (2017), p. S2-1.
3. B. Baeuerle, A. Josten, M. Eppenberger, D. Hillerkuss, and J. Leuthold, "Low-complexity real-time receiver for coherent Nyquist-FDM signals," *J. Lightw. Technol.* **36**, 5728–5737 (2018).
4. NVIDIA Corp., *NVIDIA A100 Tensor Core GPU Architecture, Whitepaper* (2020). <http://www.nvidia.com/>.
5. E. Dutisseuil *et al.*, "34 Gb/s PDM-QPSK coherent receiver using SiGe ADCs and a single FPGA for digital signal processing," in *Opt. Fiber Commun. Conf.*, (2012), p. OM3H.7.
6. L. Kull *et al.*, "A 24-to-72GS/s 8b time-interleaved SAR ADC with 2.0-to-3.3pJ/conversion and >30dB SNDR at Nyquist in 14nm CMOS FinFET," in *IEEE Int. Solid-State Circuits Conf.*, (2018), pp. 358–360.
7. J. Cao *et al.*, "A transmitter and receiver for 100Gb/s coherent networks with integrated 4x64GS/s 8b ADCs and DACs in 20nm CMOS," in *IEEE Int. Solid-State Circuits Conf.*, (2017), pp. 484–485.
8. K. Cushon, P. Larsson-Edefors, and P. Andrekson, "Low-power 400-Gbps soft-decision LDPC FEC for optical transport networks," *J. Lightw. Technol.* **34**, 4304–4311 (2016).
9. C. Fougstedt and P. Larsson-Edefors, "Energy-efficient high-throughput VLSI architectures for product-like codes," *J. Lightw. Technol.* **37**, 477–485 (2019).
10. C. Fougstedt, O. Gustafsson, C. Bae, E. Börjeson, and P. Larsson-Edefors, "ASIC design exploration for DSP and FEC of 400-Gbit/s coherent data-center interconnect receivers," in *Opt. Fiber Commun. Conf.*, (2020), p. Th2A.38.
11. C. Fougstedt, P. Johannisson, L. Svensson, and P. Larsson-Edefors, "Dynamic equalizer power dissipation optimization," in *Opt. Fiber Commun. Conf.*, (2016), p. W4A.2.
12. K. Choutagunta and J. M. Kahn, "Dynamic channel modeling for mode-division multiplexing," *J. Lightw. Technol.* **35**, 2451–2463 (2017).
13. H. Sun *et al.*, "800G DSP ASIC design using probabilistic shaping and digital sub-carrier multiplexing," *J. Lightw. Technol.* (2020). Early access, doi: 10.1109/JLT.2020.2996188.
14. K. Cushon, P. Larsson-Edefors, and P. Andrekson, "A high-throughput low-power soft bit-flipping LDPC decoder in 28 nm FD-SOI," in *European Solid-State Circuits Conf.*, (2018), pp. 102–105.
15. E. Börjeson, C. Fougstedt, and P. Larsson-Edefors, "Towards FPGA emulation of fiber-optic channels for deep-BER evaluation of DSP implementations," in *OSA Advanced Photonics Congress, SPPCom*, (2019), p. SpTh1E.4.
16. E. Börjeson and P. Larsson-Edefors, "Cycle-slip rate analysis of blind phase search DSP circuit implementations," in *Opt. Fiber Commun. Conf.*, (2020), p. M4J.3.