
# Fiber-on-Chip: Digital FPGA Emulation of Channel Impairments for Real-Time Evaluation of DSP

# Fiber-on-Chip: Digital FPGA Emulation of Channel Impairments for Real-Time Evaluation of DSP

**Per Larsson-Edefors and Erik Börjeson**

*Dept. of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden*

*perla@chalmers.se*

**Abstract:** We describe the Fiber-on-Chip (FoC) approach, in which digital models are used for real-time emulation of an optical communication system, to achieve cost-effective and reproducible long-term DSP evaluations inside a single chip. 

## 1. Introduction

Real-time optical communication experiments enable the study of DSP hardware behavior over extended periods of time, making it possible to carry out deep bit-error rate (BER) test runs during which optical transmission properties vary. In addition, since DSP power dissipation depends on switching statistics of logic signals, real-time experiments produce use-case data essential to accurate power estimation. However, real-time experiments require complex testbeds with optical/electronic components; testbeds which may include additional mechanical components to emulate time-varying properties like polarization changes [1,2]. Thus, real-time experimental approaches are significantly more complex and expensive than conventional off-line processing approaches, particularly when DSP functions are implemented in application-specific integrated circuits (ASICs) [3].

Mainly developed for coherent DSP receivers, the Fiber-on-Chip (FoC) is a test and verification concept that attempts to substitute the whole optical/mechanical testbed with on-chip digital models of the transmitter and the channel [4, 5]. Deployed in ASICs or field-programmable gate array (FPGA) circuits, the FoC enables real-time emulation using data generated on-chip. Since all digital models are under software control, it becomes straightforward to set up a number of long-term, unsupervised emulations, in which properties of the optical communication system are dynamically controlled and in which receiver performance is continuously monitored.

## 2. Fiber-on-Chip (FoC) Overview

Based on hardware descriptions (HDLs) in the VHDL language, the first FoC version [4] modeled a single-channel system with two channel impairments, viz. additive white Gaussian noise (AWGN) and phase noise, with the purpose to enable long-term evaluations of cycle slip rates in carrier phase recovery circuits [6]. Clearly, during development of an FoC system, the HDL code must be validated against a high-level system model. While HDL code has been shared under Chalmers Optical Fiber Channel Emulator (CHOICE) at www.cse.chalmers.se/research/group/vlsi/choice/, MATLAB system models are currently not shared.

As shown in Fig. 1, the receiver DSP implementation is integrated inside the FoC emulator, in which it receives a virtually endless stream of symbols from the digital channel. At the DSP receiver output, on-chip test circuits are added to monitor the receiver's function and performance, only occasionally transferring data off-chip. Some parameters are programmable in the sense that they can be changed dynamically during an emulation run, while other system properties need to be hardcoded in the implementation. Clearly an advantage of FPGAs over ASICs is their reconfigurability, making it possible to change e.g. filter lengths in between emulation runs.

Compared to MATLAB-HDL co-simulation, which is faster than fixed-point MATLAB system simulations, FPGA emulation speeds up DSP evaluation by several orders of magnitude [4]. But even if an FoC FPGA can process a large number of symbols per time unit, FoC systems that support the line-rates of industrially relevant communication systems are not feasible. This is because the resource limitation of FPGAs puts a strict limit on the degree of parallelism that can be implemented. If the emulated system is of low complexity, however, several FoC systems, each with parallel lanes, may be instantiated in parallel on an FPGA to further speed up emulations.

## 3. Digital Transmitter Models

As shown in Fig. 1, the input bitstream is generated in a random number generator (RNG), which is based on [7] and whose seed is programmable. We have previously used similar on-chip data generation schemes for FPGA and ASIC verification of forward-error control (FEC) circuits [8, 9]. In contrast, the Gaussian noise generator (GNG) used to control noise sources is based on an IP core [10], whose seed is fixed once the IP block has been created.

The digital modulator unit translates the input bitstream to BPSK, QPSK, 16QAM, 64QAM, or 256QAM symbols on an I and Q channel for two polarizations. Additionally, the input bitstream is pipelined to provide a

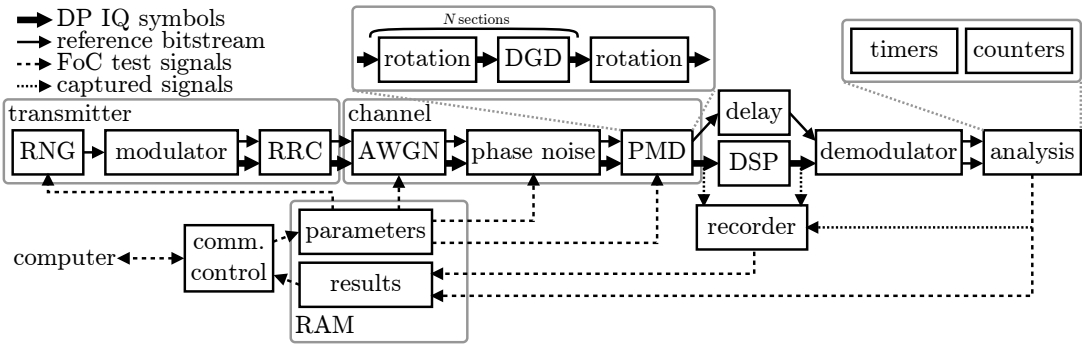

Fig. 1: FoC block diagram.

reference for test and analysis at the output. The modulated data are interpolated and shaped in the root-raised cosine (RRC) filter. Since the RRC filter uses an oversampling of two, the filter is parallelized in two data lanes. For an RRC filter with a roll-off factor of 0.1, a 51-tap FIR filter structure was used in [5]. While the filter length is defined during implementation, zeroing of programmable tap weights allows for shorter filters than implemented.

FIR tap weights can be either programmable or fixed/reconfigurable: For programmable taps, general multipliers obtain their coefficients from memories, which makes it possible to use memory addressing to dynamically change weights. But this programmability consumes logic and memory resources. In an ASIC, fixed tap weights enable hardcoded shift-and-add logic structures, which save logic area. Although FPGAs are not as customizable as ASICs, reconfigurable coefficients, which are fixed for the duration of an emulation run, save some resources. The big advantage of FPGAs is that both filter length and tap weights can be reconfigured before the next emulation.

## 4. Digital Channel Models

Digital models of three different channel impairments are currently supported: AWGN [10], phase noise [4] and polarization-mode dispersion (PMD) [5]. The GNG IP block is used for the first two impairments: The AWGN is controlled by scaling the GNG output with the signal to noise ratio (SNR) and adding the result to the symbols. The phase noise is emulated as a random walk, using GNGs scaled with the linewidth symbol-duration product. Fig. 1 shows that the PMD emulator consists of an arbitrary number of fiber sections in which differential group delay (DGD) and rotation can be set. Similar to the FIR filter example above, we can choose to implement parameters controlling DGD and rotation with fixed or programmable numbers, enabling a tradeoff between logic resource usage and channel properties which are either time varying or static.

## 5. Interfaces for Analysis and Data Transfers

In order to 1) reduce the size of the data transmitted to a computer for evaluation and 2) allow for continuous and autonomous emulation, FoC test circuits perform different types of pre-processing. This reduction is necessary, since slow FPGA-computer interfaces otherwise reduce the FoC processing speed. The test circuits include facilities to count the number of transmitted bits, detected bit-errors and cycle slips, timers to analyze e.g. the convergence time of an equalizer, and data recorders that can capture a part of any internal signal before, after, or around either a trigger pulse or a triggering event that is the result from monitoring of signals or other properties.

## 6. Example of FoC FPGA Emulation Runs

To illustrate the FoC capabilities, we have performed a series of FPGA emulation runs, where we use FoC test circuits to study the effect of varying channel and DSP parameters. These example runs are based on QPSK and use the same DSP subsystem: A 9-tap CMA-based adaptive equalizer followed by a 4th-power phase estimator. Fig. 2a and 2b show a recorded constellation diagram of the DSP input and output symbols, respectively. The SNR for this run is set to 10 dB and a 10-section PMD emulator is used. A fractional DGD of 0.05 samples/section is used for all sections and the polarization rotation is set separately for each section to a random starting value and a time-dependent rotation in the 10 to 100 krad/s range. The constellations are captured after equalizer convergence, which is detected by comparing the average equalizer error signal with a threshold value.

Fig. 2c is generated by capturing how much the bit-error counter has increased over the last 128 transmitted bits. This makes it possible to study the convergence behavior for different step-size settings in the equalizer of the DSP subsystem. Here, the PMD settings were identical to those used in Fig. 2a and 2b, but the SNR was 5 dB.

We can also study how the equalizer in the DSP subsystem responds to changes in the PMD by varying the polarization rotation rate. While having FoC continuously monitoring bit errors, we program a temporary increase in polarization rotation for four of the ten sections (1 Mrad/s for a set length of samples, called the event length). This PMD event, which takes place deep into an emulation run, leads to the recording shown in Fig. 2d.

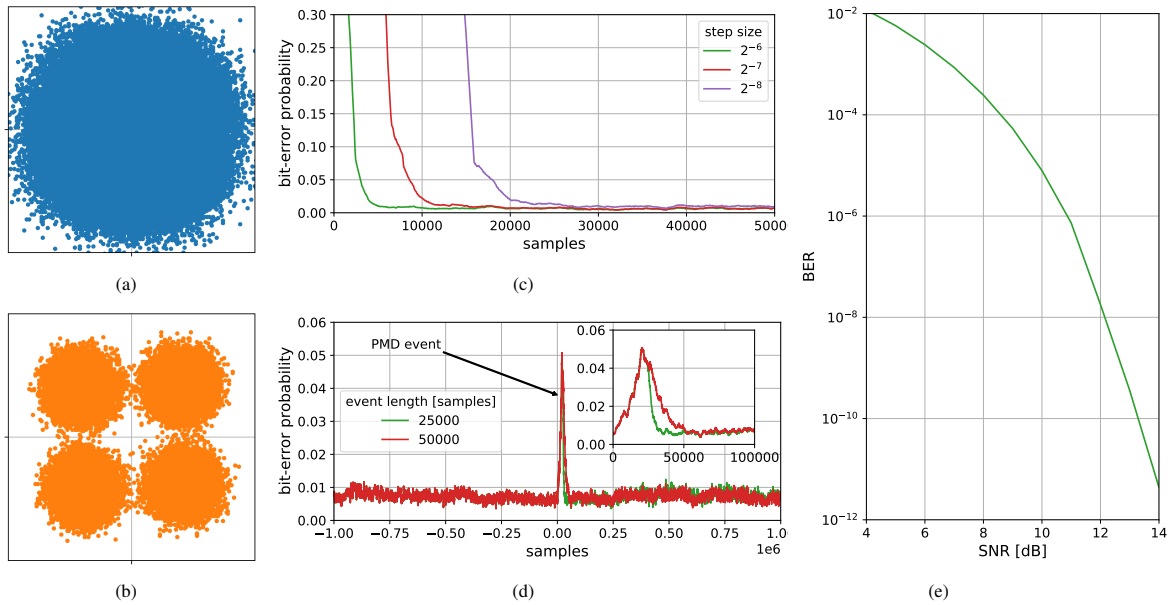


Fig. 2: Results from FoC runs, where (a) and (b) show symbol constellations before and after a DSP subsystem (see Fig. 1), (c) shows the bit-error probabilities of the convergence phase of an adaptive equalizer, (d) shows the error probabilities around a PMD event, and (c) shows a BER curve of a DSP subsystem captured after equalizer convergence. Note that both (c) and (d) have been smoothed using a moving average applied after downloading the data from the FPGA.

To show the speed-up of FoC, compared to MATLAB-HDL co-simulations, the BER of the DSP subsystem was measured with the PMD emulator turned off. For our FoC system, one hour was needed to generate the results shown in Fig. 2e. In contrast, a MATLAB-HDL co-simulation would require more than 200 days to reach the same BER level; a big gap in performance which will grow for more complex systems.

## 7. Conclusion

To facilitate real-time evaluation of DSP circuits for optical communication, the Fiber-on-Chip (FoC) approach replaces optical and electronic components with digital models implemented in an FPGA/ASIC emulator. Thanks to the emulator's processing speed, the analysis of DSP implementations becomes much faster than fixed-point simulations; thanks to the bit-accurate hardware models, reliable estimates of implementation penalties can be obtained. Since it is fully digital, FoC offers advantages such as high programmability and observability, making batches of reproducible and unsupervised long-term emulation runs possible. Among possible digital feature extensions, we may consider using pattern memories to store waveforms from optical experiments [11] making it possible to calibrate digital models to experimental data.

## References


[1] T. Pfau *et al.*, "Ultra-fast adaptive digital polarization control in a realtime coherent polarization-multiplexed QPSK receiver," in *Opt. Fiber Commun. Conf. (OFC),* (2008), p. OTuM3.

[2] S. Yamamoto *et al.*, "Hybrid 40-Gb/s and 100-Gb/s PDM-QPSK DWDM transmission using real-time DSP in field testbed," in *Natl. Fiber Opt. Eng. Conf. (NFOEC),* (2012), p. JW2A.4.

[3] 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. (OFC),* (2010), p. OMS1.

[4] E. Börjeson, C. Fougstedt, and P. Larsson-Edefors, "Towards FPGA emulation of fiber-optic channels for deep-BER evaluation of DSP implementations," in *Advanced Photonics Congress, SPPCom,* (2019), p. SpTh1E.4.

[5] H. Kan, H. Zhou, E. Börjeson, M. Karlsson, and P. Larsson-Edefors, "Digital emulation of time-varying PMD for real-time DSP evaluations," in *Asia Commun. Photonics Conf. (ACP),* (2021), p. M4H.4.

[6] E. Börjeson and P. Larsson-Edefors, "Cycle-slip rate analysis of blind phase search DSP circuit implementations," in *Opt. Fiber Commun. Conf. (OFC),* (2020), p. M4J.3.

[7] G. Marsaglia, "Xorshift RNGs," J. Stat. Softw. **8**, 1–6 (2003).

[8] K. Cushon, P. Larsson-Edefors, and P. Andrekson, "Low-power 400-Gbps soft-decision LDPC FEC for optical transport networks," IEEE J. Lightw. Technol. **34**, 4304–4311 (2016).

[9] 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.

[10] G. Liu, "OpenCores: Gaussian noise generator," (2015). https://opencores.org/projects/gng.

[11] R. Maher *et al.*, "Maximizing throughput via vertical optimization of the coherent MODEM," in *Opt. Fiber Commun. Conf. (OFC),* (2020), p. Th3E.3.