Feasibility Study of FPGA-Based Equalizer for 112-Gbit/s Optical Fiber Receivers

Fredrik Toft[†], Niclas Rousk[†], Jonas Mårtensson[‡], Marco Forzati[‡], Bengt-Erik Olsson§, and Per Larsson-Edefors[†]
[†]Chalmers University of Technology, Gothenburg, Sweden
[‡]Acreo AB, Kista, Sweden, §Ericsson AB, Ericsson Research, Mölndal, Sweden

***Abstract*—With ever increasing demands on spectral efficiency, complex modulation schemes are being introduced in fiber communication. However, these schemes are challenging to implement as they drastically increase the computational burden at the fiber receiver's end. We perform a feasibility study of implementing a 16-QAM 112-Gbit/s decision directed equalizer on a state-of-the-art FPGA platform. An FPGA offers the reconfigurability needed to allow for modulation scheme updates, however, its clock rate is limited. For this purpose, we introduce a new phase correction technique to significantly relax the delay requirement on the critical phase-recovery feedback loop.**

I. Introduction

The demand for internet capacity close to doubles each year, much due to increasing amounts of video streaming. On-off keying (OOK) modulation can be used for 10 Gbit/s (to some extent even for 40 Gbit/s) fiber optic communication, but such a simple modulation scheme does not fit contexts with higher transmission rates [1]. Replacing a simple modulation scheme like OOK with a more complex one, for example, quadrature phase-shift keying (QPSK) or, even more so, 16 quadrature amplitude modulation (16-QAM), leads to a drastic increase in the computational demand on the subsequent DSP baseband circuitry. An ASIC-based DSP solution can provide the necessary performance [2], however, it also brings two disadvantages: Beside being extremely costly to develop, once it has been manufactured, an ASIC cannot be modified to accept variations of the modulation scheme.

Inspired by modulation schemes for RF systems, this paper explores implementation aspects of receivers for above 100 Gbit/s fiber optic systems [3]. Beside the goals of high performance and power-efficient operation, the implementation should allow for upgrades even after the system has been deployed. Thus, this paper evaluates if reconfigurable FPGA technology can be used for such a system. Specifically we address the equalizer, as this contains FIR filters that represent a large part of the system's footprint and, secondly, a phase-recovery loop that represents a timing-critical part of the system.

II. System Overview

To reach 100 Gbit/s, complex modulation schemes are needed; for example, QPSK carries two bits/symbol, while 16-QAM carries four bits/symbol. These schemes bring the obvious advantage of higher transmission rates or lower baud rates. However, since the distance between each symbol in the constellation is reduced, a higher signal-to-noise ratio is required.

A 100-Gbit/s 16-QAM prototype system is shown in Fig. 1. Data generated using a 7-Gbit/s PRBS-generator are converted to a binary phase-shift keying signal, having two bands with center frequencies of 14 and 24 GHz, which in turn is converted to a QPSK signal, via splitting and phase shifts. The QPSK signal is further split in two, delayed for decorrelation, amplified and added together in phase [1]. Both channels are filtered through bandpass filters before they are added together into the final signal. In the optical transmission, the signal is divided into two polarizations, delayed for decorrelation, added together and then transmitted through a 824-km optical fiber. On the receiver side, the signal is amplified and filtered through an optical bandpass filter. A local oscillator (LO) laser is used as coherent detection of both polarizations, the optical signal is converted to electricity by two balanced photodetectors (BPD) and transferred via an analog-to-digital converter (ADC) to the digital domain. In the current prototype system setup, the ADC is represented by a 50 Gsample/s oscilloscope with a resolution of 8 bits and a bandwidth of 16 GHz, of which 4 GHz is used to sample the signals.

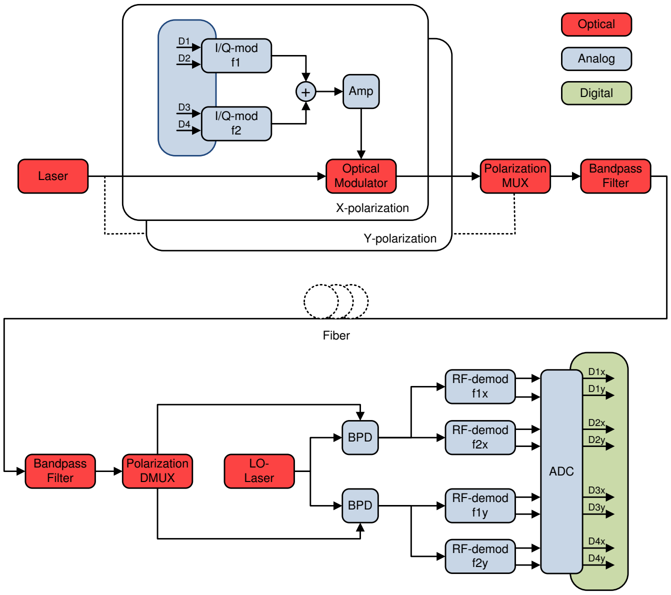


Fig. 1. Overview of system.

The oscilloscope data are sent for post processing in MATLAB. As shown in Fig. 2, the two frequency bands are separated by demodulating each channel from 14 and 24 GHz down to baseband. Chromatic dispersion (CD) is compensated for in an FIR filter in the frequency domain; this is done in a static filter to remove most of the effect [4]. The residual CD is removed later, in adaptive filters [1]. The carrier frequency recovery starts by taking the FFT of the data raised to the power of four to move the carrier frequency outside the spectrum of the data [5]. As the sample rate is 50 Gsample/s and the baud rate only 7 Gsample/s, the signal needs to be resampled. By sorting the samples into batches and aligning them based on phase, it is possible to extract a synchronous two samples/symbol signal [6].

The equalizer in Fig. 2 is based on a butterfly structure FIR filter for polarization demultiplexing and a phase-recovery loop for phase-noise cancelation. To initialize the system, the constant modulus algorithm (CMA) is used for a first estimation of FIR filter

978-1-4673-0219-7/12/$31.00 ©2012 IEEE

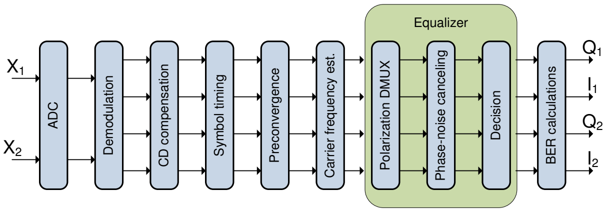


Fig. 2. Flow for signal processing.

constants [7], [8]. The phase-noise compensation is calculated by dividing the sample by the value from the decision process to generate an estimation of the error. The last step in the signal processing flow is to calculate a bit-error rate (BER) value for evaluation purposes.

After the decision mapping to the 16-QAM constellation, errors remain in the data. Forward error correction (FEC) algorithms can rectify the erroneous data, given that the BER value is limited. For example, with the continuously interleaved BCH enhanced FEC (CI-BCH eFEC) it is possible to improve BER from $4.6 \cdot 10^{-3}$ to 10^{-15} [9], at an overhead of only 6.7%. Since the FEC unit is separate from the preceding circuitry, the equalizer's output is specified to stay below a BER value of $3 \cdot 10^{-3}$.

III. Equalizer Algorithm

The adaptive equalizer algorithm (Fig. 3) uses four adaptive FIR filters in a butterfly structure. The input to the four FIR filters is the received I/Q data corresponding to the two different polarizations.

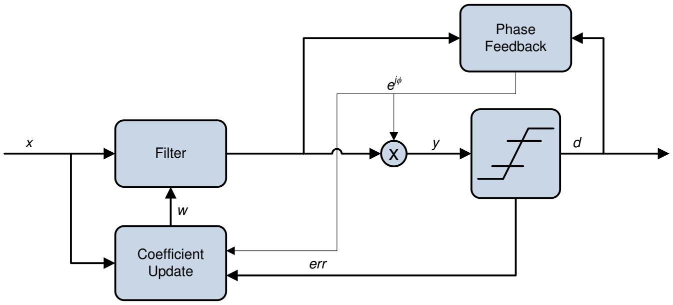


Fig. 3. Equalizer algorithm.

The task of the equalizer is to perform polarization demultiplexing of the signal and to mitigate any linear impairment such as polarization-mode dispersion (PMD), polarization-dependent loss (PDL) and residual chromatic dispersion [1]. The FIR filter takes two samples per symbol as input and outputs one sample per symbol, hence performing an effective downsampling of two.

A. FIR Filter

The FIR filter is updated using an LMS algorithm to find the coefficients that will produce the smallest error between the desired signal and the actual signal [1]:

$$w_{mn}^{k+1} = w_{mn}^{k} + \mu \varepsilon_m x_n, \quad m = 1,2; \quad n = 1,2; \tag{1}$$

where x_n is the input to the FIR filter, μ is the equalizer step size, w_{mn}^{k} are the old filter coefficients and w_{mn}^{k+1} are the new filter coefficients. The subscript index m and n corresponds to the two polarizations, respectively, and index k represents the sample number.

The cost function ε used in the equalizer is the difference between the actual output y^k and the desired output d^k. A rotation by ϕ^k is also applied to the LMS error in order to decouple the adaptation of the equalizer's taps from the phase noise tracking. The error is calculated as:

$$\varepsilon_m = e^{j\phi_k}(d_m^k - y_m^k) \tag{2}$$

B. Phase-Recovery Loop

Before taking the decision on which symbol the output of the FIR filter corresponds to, varying frequency offset and laser phase noise must be tracked and compensated for. The phase error is estimated from the N previous samples and subtracted from sample k and polarization m as [1]:

$$\phi_m^k = \arg\left(\sum_{i=1}^{N} y^{k-i}/d^{k-i}\right) \tag{3}$$

The rotation ϕ_m^k is fed back and applied after the FIR filter, before the decision function, by multiplying the filter output with $e^{-j\phi_m^k}$.

C. Decision

The decision part of the system is a simple level comparator that maps the filtered and phase noise compensated output y_m^k to a symbol in the 16-QAM constellation. Gray mapping of the constellation diagram is used to improve the BER value, since the two closest points to each symbol only differ by one bit.

IV. System Resource Allocation

FPGAs in advanced process technologies offer a massive amount of computational power that can be (re)configured to take on parallel and pipelined algorithms. The one-size-fits-all computational structure, however, prohibits FPGAs to reach as high performance and low power as ASICs. Assuming a state-of-the-art FPGA—a 28-nm Virtex 7 in advance product specification DS180 v1.5 [10]—an estimate of the required parallelism can be made. The maximal clock frequency is 638 MHz [10], but we will use a conservative value of 500 MHz. The maximal bit rate is 500 MHz $\cdot$ 2 frequency channels $\cdot$ 2 polarizations $\cdot$ 4 bits per sample (16-QAM) = 8 Gbit/s. For a throughput of 112 Gbit/s, the system has to calculate at least 14 samples in parallel.

The number of filter taps determines how many multiplications that needs to be performed for each sample. Given two complex numbers $A + jB$ and $C + jD$, a complex multiplication is done using three multipliers and four adders according to:

$$\begin{aligned} \Re[(A+jB)(C+jD)] &= AC - BD \\ \Im[(A+jB)(C+jD)] &= (A+B)(C+D) - AC - BD \end{aligned}$$

We can now estimate the number of filter taps that are possible to implement. Assuming three multipliers per complex multiplication, two polarizations, two FIR filters per polarization, and 14 parallel calculations on one of the the most powerful models[1] at most $3,960/(3 \cdot 2 \cdot 2 \cdot 14) \approx 24$ taps can be supported.

Reducing the filter tap count below this limit is beneficial, as a less complex FPGA may be used. Consequently, a number of early simulations are carried out to ensure that the constraint on BER values can be accommodated with the limited tap count. As a result, the indication is that 21 filter taps are sufficient.

V. Implementation

Using a bottom-up approach in MATLAB, we evaluate changes to the system, for example, by studying relative BER values, before committing portions of the equalizer algorithm to VHDL. The data initially used during development was the noisiest data available; upon successful evaluation, the noisy data is followed by a range of different data sets.

[1] XC7VX850T is specified to have 3,960 DSP48 slices [10].

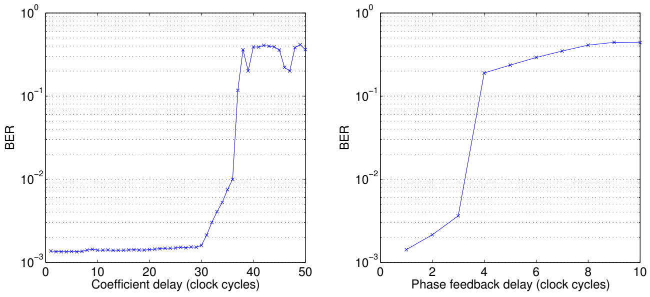
A. Number Representation

In MATLAB, by default all numbers are represented in 64-bit double-precision floating-point format. While it is possible to perform floating-point operations in an FPGA, using a fixed-point representation is preferable in terms of hardware efficiency. Using the MATLAB “Fixed-point toolbox”, a fixed-point system model was developed, taking into consideration FPGA resource constraints, such as bitwidth limitations for DSP48 input operands.

System simulations using fixed-point number representations were carried out in MATLAB, assuming a smoothing factor $N = 40$. Here, the FIR inputs were made 11 bits wide, the FIR outputs were 19 bits wide, and the coefficients were 8 bits wide. In going from floating-point to fixed-point representations, for a system with a coefficient update delay and phase-recovery delay of 20 and 3 cycles, respectively, the BER increased from $2.8 \cdot 10^{-3}$ to $3.2 \cdot 10^{-3}$.

B. Feedback Loops

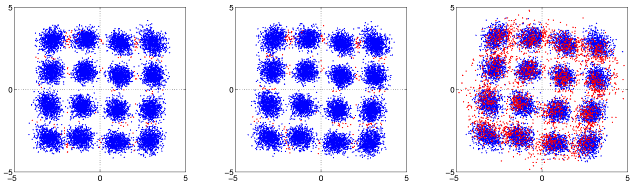
The update of the filter coefficients is done with some delay that results from parallelism as well as pipelining. The total delay in samples can be expressed as $p \cdot n$, where p is the number of parallel calculations each clock cycle and n is the number of cycles needed for the calculations. Assuming $p = 14$ and 21 filter taps, the impact of loop delays on BER is shown in Fig. 4.



(a) Coefficient (b) Phase-recovery loop

Fig. 4. Impact of loop delays on BER.

As shown in Fig. 4(b), the phase-recovery loop is very sensitive to delays; the system can tolerate approximately three clock cycles of delay, but no more. Fig. 5 shows how the constellation gets increasingly rotated as the feedback delay is increased, mapping samples to the wrong symbol, increasing BER.

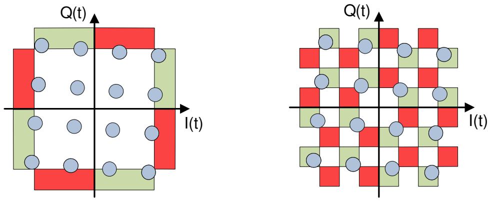


(a) 1-cycle delay (b) 2-cycle delay (c) 4-cycle delay

Fig. 5. Impact of phase-recovery loop delay on 16-QAM constellation. Red dots represent samples that are mapped to the wrong symbol.

In the phase-recovery loop, the phase error ϕ^k (Eq. 3) needs to be calculated as well as the complex exponential $e^{-j\phi^k}$. These computations cannot be completed within three clock cycles even in state-of-the-art FPGA technology.

In order to relax the strict timing requirements on the phase-recovery loop, a technique for tracking and compensating the phase rotation was employed; see Fig. 6(a). A counter counts samples that get mapped to specific pre-defined areas on the graph. Samples that get mapped to a green area correspond to a clock-wise rotation, whereas samples mapped to a red area correspond to a counter clock-wise rotation.



(a) Coarse (b) Fine

Fig. 6. Rotation tracking techniques: (a) coarse-grain tracking, and (b) fine-grain tracking.

When a sample is mapped to a green and a red area, the counter is decremented and incremented, respectively. The counter value c is then divided by a constant a and subtracted from the phase error ϕ^k:

$$\phi^k_{m,new} = \phi^k_m - \frac{c}{a} \qquad (4)$$

The ideal value of a is when the ratio $\frac{c}{a}$ corresponds to the rotation of the constellation for sample k. The constant a depends on the limits of the red and green areas and is decided empirically.

The rotation tracking can track the rotation of each point in the constellation, instead of the whole constellation, similar to the phase-locked loop used in earlier radio systems [11]; see Fig. 6(b). Fig. 7 shows the impact on the BER value when the techniques above are implemented. The counter is delayed as long as the phase, before it is added to ϕ^k according to Eq. 4. The constant a was set to 512 and 1024 for the coarse-grained and fine-grained technique, respectively. Using a rotation tracking counter allows for having approximately 15 clock cycles of delay for a BER of $3 \cdot 10^{-3}$.

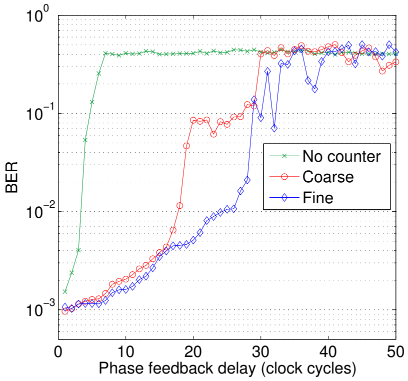


Fig. 7. Impact of rotation counter on BER.

C. Other Implementation Issues

Complex multipliers are created using the Xilinx CORE-gen and “Complex multiplication 4.0” [12], which allow the compiler to effectively map multiplications to DSP48 [13]. To reduce the calculation effort, the downsampling of two is integrated into the FIR filter by calculating every other sample. To save area, all the parallel filters

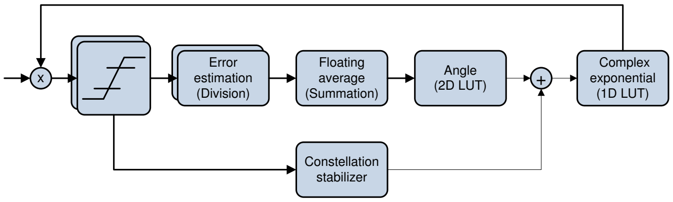

x
Error estimation (Division)
Floating average (Summation)
Angle (2D LUT)
+
Complex exponential (1D LUT)
Constellation stabilizer

Fig. 8. Hardware view of phase feedback.

are updated with the same coefficients, using Eqs 1 and 2. By setting μ to a power of two, straightforward shifting can be used.

In the phase-recovery loop (Fig. 8), an estimation of the phase error (err_k) is obtained by dividing the input data y^{k-i} with the decided one d^{k-i} (Eq. 5). Since the values to which the constellation maps are known, their inverses can be precomputed, enabling straightforward complex multiplication instead.

$$err_k = e^{j \cdot \arg\left(\sum_{i=1}^{N} y^{k-i}/d^{k-i}\right)} \tag{5}$$

A floating average is calculated by summing the incoming values and the N previous sums. The argument calculation in the phase-recovery loop is based on a two-dimensional LUT. The sign is removed from both I and Q, which saves two bits in the LUT (these are restored after the exponential), saving significant area but increasing latency somewhat. The rotation tracking counter sums hits for clockwise and counter-clockwise rotation and stores the difference. This difference is then divided by a power-of-two scaling factor and added to the calculated angle (ϕ^k).

VI. Evaluation

MATLAB simulations were performed for the system in Sec. V using the bitwidths of Sec. V-A, but with a smoothing factor that can be reduced to 28 (a multiple of the parallelization of 14) thanks to the extended phase feedback delay. Using 21 taps, a coefficient delay of 20, a phase feedback delay of 12, the average BER becomes $2.2 \cdot 10^{-3}$ for a number of batches of 500,000 samples from the 5 million sample test data that resulted from the setup in Sec. II. Assuming an XC7VX850T model with 3,960 DSP48 and 133,500 logic blocks [10], the system requires the resources shown in Table I.

TABLE I
Resource Utilization on XC7VX850T

Design part	#DSP48s	#Logic slices
FIR filters	3,528	37,400
Phase recovery	128	2,200
LMS update	0	49,500
Total	3,656 (92%)	89,100 (67%)

The VHDL implementation was also simulated to obtain active power dissipation values. We generated a switching activity interchange format (SAIF) file for one of the fourteen 21-tap FIR filters, by using a use case comprising 200 input test vectors from MATLAB simulations. The power dissipation of one FIR filter running at 500 MHz was 1.05 W, making the entire parallel FIR filter dissipate 14.7 W. The default settings of the FPGA tools were used, that is, constant toggle rates of 50% were assigned to all nodes, for other blocks. Considering the overestimation made in this method, the power dissipation of the LMS update and the phase-recovery blocks is estimated to be 6 W. In total, the power dissipation of the current system implementation is approximately 20 W.

VII. Conclusion

In this feasibility study, we have shown that a 16-QAM 112-Gbit/s decision directed equalizer for a fiber optical receiver can be implemented on a state-of-the-art FPGA. Our MATLAB simulations showed that our system setup, which uses 14 parallel FIR filter lanes, each with 21 taps, gives a BER of $2.2 \cdot 10^{-3}$.

In the implemented system, a number of important design trade-offs have been made: Bitwidths for fixed point circuitry need to be established, arithmetic operations have to be mapped carefully to FPGA resources, and feedback delays must be analyzed with FPGA properties in mind. The system implementation, however, has not been optimized for power dissipation in any way. The LMS update in particular can be refined by doing fewer updates to the coefficients.

The feedback delay of the phase-recovery loop proved to be too strict in the initial equalizer algorithm, and it is only thanks to the developed rotation tracking counter that the phase correction of the equalizer can be implemented. As the entire DSP chain of the complete receiver contains several feedback loops, features of the rotation tracking algorithm employed inside the equalizer may be introduced in other parts of the system.

VIII. Acknowledgement

We wish to thank Dr. I. Sourdis and Dr. L. Svensson for their involvements in technical discussions.

References

[1] B.-E. Olsson *et al.*, "RF-Assisted Optical Dual-Carrier 112 Gbit/s Polarization-Multiplexed 16-QAM Transmitter," in *Optical Fiber Communication Conf. and Exposition (OFC/NFOEC)*, March 2010.

[2] S. Chandrasekhar and X. Liu, "Enabling Components for Future High-Speed Coherent Communication Systems," in *Optical Fiber Communication Conf. and Exposition (OFC/NFOEC)*, March 2011.

[3] T. Pfau *et al.*, "Hardware-Efficient Coherent Digital Receiver Concept With Feedforward Carrier Recovery for M-QAM Constellations," *J. Lightwave Technology*, vol. 27, no. 8, pp. 989–999, April 2009.

[4] S. J. Savory, "Digital Filters for Coherent Optical Receivers," *Opt. Express*, vol. 16, no. 2, pp. 804–817, January 2008.

[5] S. J. Savory *et al.*, "Electronic Compensation of Chromatic Dispersion Using a Digital Coherent Receiver," *Opt. Express*, vol. 15, no. 5, pp. 2120–2126, March 2007.

[6] C. R. S. Fludger *et al.*, "Coherent Equalization and POLMUX-RZ-DQPSK for Robust 100-GE Transmission," *J. Lightwave Technology*, vol. 26, no. 1, pp. 64–72, January 2008.

[7] P. Winzer and A. Gnauck, "112-Gb/s Polarization-Multiplexed 16-QAM on a 25-GHz WDM Grid," in *34th European Conf. Optical Communication*, 2008.

[8] I. Fatadin *et al.*, "Blind Equalization and Carrier Phase Recovery in a 16-QAM Optical Coherent System," *J. Lightwave Technology*, vol. 27, no. 15, pp. 3042–3049, August 2009.

[9] M. Scholten *et al.*, "Enhanced FEC for 40G / 100G," Vitesse, Tech. Rep., 2009.

[10] *7 Series FPGAs Overview, DS180 (v1.5), Xilinx*, Mar. 2011. [Online]. Available: http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf

[11] A. Mouaki Benani and F. Gagnon, "Comparison of Carrier Recovery Techniques in M-QAM Digital Communication Systems," in *Canadian Conf. Electrical and Computer Engineering*, 2000, pp. 73–77, vol. 1.

[12] *LogiCORE IP Complex Multiplier, v4.0, Xilinx*, Mar. 2011. [Online]. Available: http://www.xilinx.com/support/documentation/ip_documentation/ds793_cmpy.pdf

[13] *7 Series DSP48E1 Slice, UG479 (v1.2), Xilinx*, Oct. 2011. [Online]. Available: http://www.xilinx.com/support/documentation/user_guides/ug479_7Series_DSP48E1.pdf