Real-time Implementation of Digital Coherent Detection

R. Noé, U. Rückert, S. Hoffmann, R. Peveling, T. Pfau, M. El-Darawy, A. Al-Bermani

University of Paderborn, EIM-E, Warburger Str. 100, 33098 Paderborn, Germany, noe@upb.de

Abstract *The implementation of algorithms for coherent detection of advanced modulation formats imposes constraints. A hardware-efficient phase estimator is presented, and measurement results with a CMOS receiver chip designed for 40 Gb/s digital coherent polarization-multiplexed QPSK.*

Introduction

Coherent detection with digital signal processing enables optical transmission systems with 100 Gb/s data rate or above Advanced modulation formats such as quadrature phase shift keying (QPSK), multi-level quadrature amplitude modulation (QAM) and orthogonal frequency division multiplexing (OFDM) in combination with polarization multiplex can be employed on such systems. They offer high spectral efficiency and comparatively high tolerance against chromatic dispersion (CD) and polarization-mode dispersion (PMD). These fiber impairments can be mitigated in the electronic digital domain.

The phase and frequency offsets between signal and local oscillator laser can be tracked, even if DFB lasers are employed[1]. Polarization crosstalk, CD and PMD can be compensated and also nonlinear phase noise can be mitigated[2]. Several realtime experiments using such algorithms have already been demonstrated[3,4]. But due to the high complexity in the implementation of realtime coherent receivers, offline digital signal processing is used in most experiments[5,6]. It is a very useful way to investigate new algorithms without the need to do extensive hardware development. Experimental results with a true realtime measurement setup have been published[7,8] but are still rare to find.

The starting point of this paper is therefore to analyze the general constraints for realtime signal processing unit (DSPU) algorithms to ensure the usability in real systems. As an example for hardware efficiency, a QPSK phase estimation algorithm is presented. Finally, experimental results with an integrated DSPU circuit for a 40 Gb/s realtime coherent QPSK receiver are presented. Phase estimator and the DSPU circuit were developed by the Univ. Paderborn in the ”synQPSK” project funded by the European Commission.

Realtime constraints for receiver algorithms

The high data rates in optical communication of 43 Gb/s, 112 Gb/s or even above generate stringent constraints for the algorithms suitable for realtime coherent receivers (Fig. 1).

The receiver consists of an optical frontend including optical 90° hybrids, O/E conversion, analog-to-digital converters (ADCs) and a DSPU. ADCs and DSPU can either be integrated in a single chip to ease interfacing and reduce the footprint[4], or in a modular approach for optimized performance, where ADCs and DSPU can be developed in different technologies for maximum bandwidth and high integration, respectively[9]. In both topologies the implemented algorithms must allow parallel processing as shown in Fig. 2.

The DSPU cannot operate directly at the sampling clock frequency of the analog-to-digital converter, which is in general 10 GHz or higher, but requires demultiplexing to process the data in m parallel modules at clock frequencies below 1 GHz. This allows automated generation of the layout, which is indispensable due to the complexity of the system. A comparison between the sampling clock frequency and the divided clock shows that at least $m \geq 16$ parallel modules are required. Algorithms for realtime applications should therefore theoretically allow parallel processing with an unlimited number of demultiplexed channels. The requirement for this is that the output of one module is independent of the outputs of the other parallel modules.

A good example is the comparison of two filter structures: Finite impulse response (FIR) filters and infinite impulse response (IIR) filters. Fig. 3 depicts both structures in both serial and parallel systems. It can be seen that it is easily possible to parallelize an FIR filter. Though neighbouring modules depend partly on the same inputs, they do not depend on the output signal of another parallel module. In contrast it is impossible to realize in practice the parallel structure shown for the IIR filter, because the output

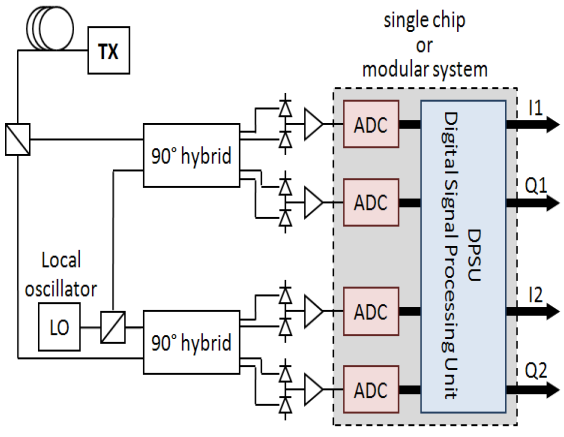

Fig. 1: Coherent optical receiver structure

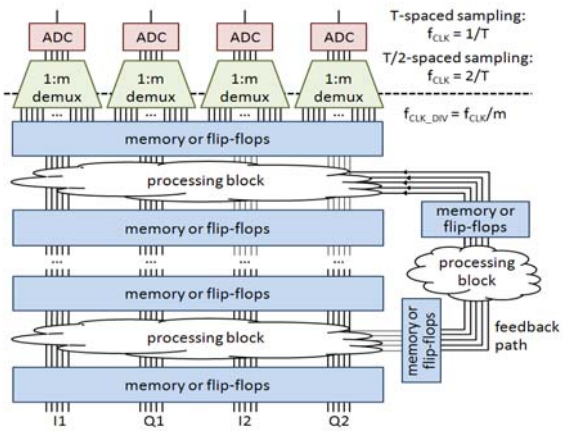

Fig. 2: Internal structure of the DSPU

978-3-8007-3173-2

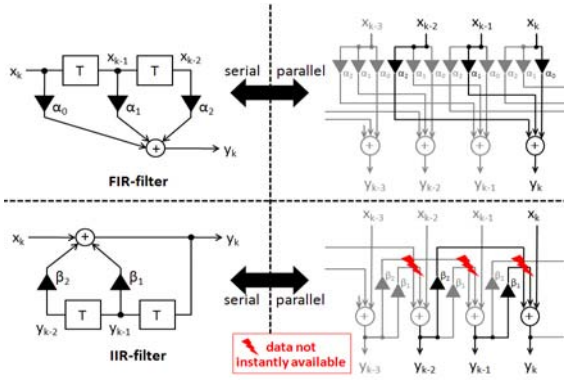

Fig. 3: Serial and parallel implementation of the DSPU

of the each module depends on the outputs of all previous parallel modules. An extremely low clock frequency would be needed to allow all calculations to be executed within one clock cycle. This would result in an even higher number of parallel modules, which makes an implementation unfeasible.

Another important consideration for algorithms to be suitable for realtime applications is the tolerable feedback delay. In simulation or offline processing, feedback delays of 1 symbol are easy to achieve, but this is impossible in a realtime system. The reasons are the parallel processing and massive pipelining, which is required to cope with the high data rates. Pipelining means that only fractions of the whole algorithm are processed within one clock cycle and the intermediate results are stored in buffers (e.g. memory or flip-flops) as shown in Fig. 2. Therefore it can take easily >100 symbol durations (~10 ns @ 10 Gbaud), until a received symbol has an impact on the feedback signal. For polarization control or CD/PMD compensation, which in general use integral controllers with time constants in the µs-range[3], the several ns additional delay due to pipelining can be neglected. But the feedback delay can have a severe impact on the performance of algorithms that require an instantaneous feedback, e.g. decision-directed carrier recovery, which is often used in offline signal processing for higher-order QAM[6] .

Finally, another important constraint for realtime coherent receiver algorithms is hardware efficiency, which also originates from the parallel processing in the DSPU. Because most of the required algorithm blocks have to be implemented *m* times within the DSPU, computationally intensive algorithms require a huge amount of chip area and therefore increase power consumption and cost. The implemented algorithms should therefore not only be evaluated by performance, but also by hardware efficiency.

An example for such a hardware-efficient design is the angle-based phase estimator for QPSK presented in the following section[10]. It replaces the usual phase estimator of Viterbi and Viterbi[11] that requires several coordinate conversions and complex calculations. To simplify notation, the algorithm is explained for a single polarization system first. It can be easily extended to a polarization multiplex system, for which a CMOS implemetation is presented afterwards.

Angle-based phase estimation

Consider a single-polarization transmission system. After optoelectronic conversion the discretized received symbols can be described as a sequence of complex numbers $Z(k)$ with the argument $\psi(k)$ and time index k. For phase estimation and data recovery, only $\psi(k)$ is necessary. Nevertheless, the usual way to estimate phase involves complex calculations[1,11], although the magnitudes $|Z(k)|$ merely contain fluctuations due to noise

Usually, the sent QPSK symbols are differentially encoded, which allows to build a simple asynchronuos receiver. Instead of multiplying the received complex symbol with its complex conjugate predecessor, it is sufficient to calculate the difference $\psi(k)-\psi(k-1)$. This difference is finally used in a simple decision circuit that recovers the data.

For a synchronous receiver, an estimated phase angle $\hat{\varphi}(k)$ is subtracted from each angle in order to compensate for phase noise and residual frequency mismatch. Best results for high phase noise requirements can be achieved with estimators that perform a full phase tracking.

The estimated phase is usually limited to a certain quadrant, therefore it is wrapped. Phase unwrapping is feasible in realtime, but it is easier to detect the deviations between physical and wrapped estimated phases and to consider them in the differential decoder that recovers the original data[1].

The received symbol $Z(k)$ consists of the sent QPSK symbols $c(k)=\pm 1\pm j$, a time-variant phasor $e^{j\psi(k)}$ and additional noise $n(k)$,

$$Z(k)=|Z(k)|\cdot e^{j\psi(k)}=c(k)\cdot e^{j\varphi(k)}+n(k). \qquad (1)$$

Because $c^4(k)=-4$ is constant, the fourth power $Z^4(k)$ is almost independent of QPSK modulation for small noise contributions. Viterbi and Viterbi (V&V) generalized this modulation removal approach in polar coordinates to the product of two complex functions[11]. The result is

$$X(k)=|Z(k)|^u\cdot e^{j4\psi(k)} \quad (u=0,2,4) \qquad (2)$$

As a replacement of $Z^4(k)$, the complex signal $X(k)$ can be averaged in order to remove the noise, and finally an estimated phase can be obtained from the argument of the complex filter output.

The V&V phase estimator converts the input signal into polar coordinates, performs nonlinear functions of its magnitude and phase and generates a complex signal $X(k)$. In order to enable complex filtering, $X(k)$ is represented in cartesian coordinates but it is finally converted into polar coordinates again. In order to avoid multiple coordinate conversions, an angle-based approach called barycenter algorithm can employ the position angle of the received symbol within its quadrant which is defined as

$$\vartheta(k) = \psi(k) \bmod \frac{\pi}{2}. \quad (3)$$

The position angle $\vartheta(k)$ is independent of QPSK modulation which makes it useful for phase estimation. Two position angles allow to calculate immediately an average phase angle $\mu = f(\vartheta(j), \vartheta(k))$ without complex calculations, in order to imitate V&V phase estimator behavior with $u = 0$ (normalized magnitudes). The angle average results can be employed for further pair averaging elements called cells or nodes. Note that, compared to V&V phase estimation, only the magnitude of complex intermediate results is neglected. The barycenter algorithm[12] is therefore equivalent to V&V estimation with additional normalization of each partial sum.

Similiar to a complex adder tree required for V&V phase estimators, it consists of equal elements that calculate partial results from two input values. Parallelization and pipelining enable fast and hardware-efficient calculation.

In the angle-based approach, all input, output and intermediate values are angles. The nodes convert pairs of position angles α, β into average position angles μ. The partial result from each node is calculated within one DSPU cycle using the sum $\sigma = \alpha + \beta$ and the difference $\delta = \alpha - \beta$ of the input values for the final calculation, thus avoiding the complex calculation of the V&V estimator,

$$\mu = \left(\frac{\sigma}{2} + \frac{\pi}{4} \left\lceil \frac{|\delta|}{\pi/4} \right\rceil \right) \bmod \frac{\pi}{2}. \quad (4)$$

The node interpolates the phase track on the shortest possible path which is supposed to be most likely. Therefore, the barycenter algorithm[12] is a maximum likelihood approach.

Compared to a complex summation, the magnitude of the intermediate result is not calculated, only its phase. But this magnitude would be necessary for an angle-based equivalent of the V&V estimator. But on the other hand, perfect equivalence is not required, only equal performance. Magnitude of the intermediate result could be calculated from δ with a cosine function, but it turned out in simulation[13] that it is sufficient to convert each δ into a single reliability bit by a simple comparison.

The reliability bits from two nodes are employed by a Boolean selectivity mechanism in the subsequent node. Averaged angles marked as not reliable are only used if there is no choice, i. e. both input values are marked as not reliable. For the new intermediate result, a reliability bit is inherited or generated. Two input values initially marked as not reliable can produce a reliable output. The selection process based on the reliability information leads to the name selective maximum likelihood phase approximation (SMLPA) for the modified barycenter algorithm.

Integrated DSP for a coherent QPSK receiver

Fig.4 shows the chip micrograph of the fabricated DSPU chip. The C-shaped structure on the left is a full custom design consisting of a clock divider (center, left) and demultiplexers for both polarizations (top, bottom). The standard cell DSPU part is the T-shaped structure on the right.

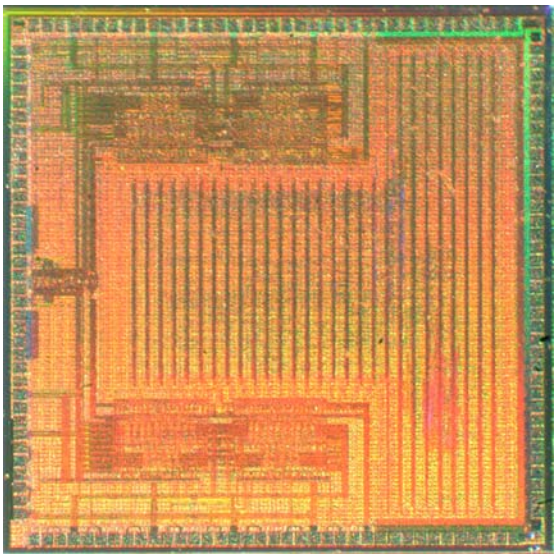

Fig. 4: Chip micrograph of the DSPU chip

The DSPU unit contains polarization control, phase estimation and data recovery for a polarization-multiplexed QPSK receiver. For polarization control, a decision directed correlation based approach was chosen[1]. This algorithm imposes some feedback delay on the system but avoids the transmission of a special training sequence for the polarization control. Based on the assumption that received symbols after decision are in most cases identical with the transmitted symbols, the algorithm can blindly minimize crosstalk and polarization dependent loss (PDL).

After demultiplexing into 16 parallel modules, the four 5 bit input vectors multiply a complex matrix *M* to compensate for polarization crosstalk and PDL. The matrix multiplication is followed by the feedforward carrier & data recovery that uses an angle-based approach presented above. For each pair of complex received and compensated symbols, a pair of angles is determined. A common phase track is estimated for both polarizations.

The polarization control matrix is updated in a decision-directed approach by correlating the data before and behind the decision circuit in one out of 16 parallel modules. The polarization control time constant can be switched between c_1 = 0.4 µs and c_2 = 1.6 µs (@ 10 Gbaud) by changing the control gain between $g_1 = 2^{-8}$ and $g_2 = 2^{-10}$ through an asynchronous serial debug interface. Together with this interface also other test and debug structures like a built-in self-test or BER counters are included on the chip to ease testing and configuration. The chip has a total complexity of 1.23M transistors in the standard-cell part, 12k transistors in the full-custom part and a power consumption of ~2 W.

Test results of the receiver chip

The DSPU CMOS chip depicted in Fig. 4 was tested in a realtime experiment together with four equal analog digital converters (ADCs) developed in our group[14]. The target system of the synQPSK project

was a QPSK polarization multiplex transmission system with a data rate of 40 Gbit. Therefore the ADCs had to cope with a symbol rate of 10 GBaud. The 5 bit 12.5 Gs/s ADCs were fabricated in a SiGe technology in a pure flash topology without track-and-hold amplifier. After binary encoding, the output signals are fed into the CMOS chip input stages over differential transmission lines on a ceramic board. Input stages of the CMOS chip employ full custom source coupled logic (SCL) design[15].

Although the ADCs and the CMOS chip alone were able to operate at the target frequency, this chip-to-chip interfacing turned out to be a major drawback of the chosen modular system because of insufficient signal quality. Imperfect suppression of switching noise was identified as the reason.

The presented transmission experiments have therefore been performed at a reduced symbol rate of 2.5 GBaud. Speed calculations for the polarization control are downscaled accordingly.

Fig. 5 shows the achieved bit error rates for three different polarization control cases against the input power of the preamplifier. The bit error rates of all output channels are averaged. For the measurements without polarization crosstalk the best performance was achieved with a BER floor of $1.3 \cdot 10^{-5}$ and a sensitivity of -44.1 dBm for a BER of 10^{-3}. The measurements with 50% polarization crosstalk suffers from higher penalties with a BER floor of $5.4 \cdot 10^{-5}$ and a sensitivity of −41.4 dBm for a BER of 10^{-3}. The results for polarization scrambling with 50 rad/s lie in the middle with a BER floor of $3.3 \cdot 10^{-5}$ and a sensitivity of -42.6 dBm.

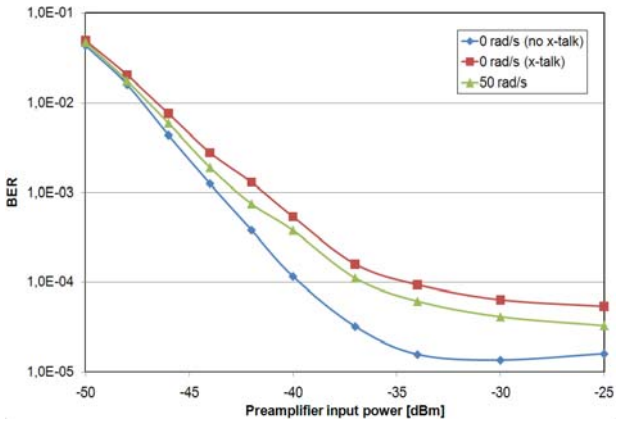

Fig. 5: Bit error rates vs. preamplifier input power for transmission at 10 Gb/s for different polarization states at the receiver input.

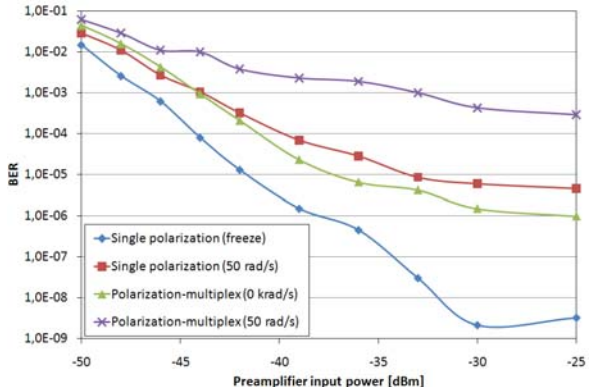

Fig. 6: Bit error rate vs preamplifier input power for transmission with and without polarization multiplex.

Fig. 6 shows the achieved BER vs. input power of the preamplifier at a symbol rate of 2.5 Gbaud with and without polarization multiplex. This also shows the effect of the switching noise on performance. If the polarization control is bypassed (freeze), which deactivates roughly half of the cells in the DSPU, BER performance is superior to the case when the whole DSPU is active (polarization-multiplex, 50 rad/s). For the case of polarization mutliplex with 0 rad/s and for single polarization transmission with polarization scrambling at 50 rad/s roughly 3/4 of the DSPU cells are active.

With an FPGA-based implementation optimized for speed we have also tracked polarization fluctuations of 40 krad/s, combined with a polarization-dependent loss of 6 dB[16].

Conclusions and Outlook

We have derived three basic constraints for algorithm development for realtime digital coherent receivers: Feasibility of parallel processing, hardware efficiency consideration of feedback delays and hardware efficiency. Additionally, we have presented an angle-based phase estimator and measurement results of a CMOS DSPU for a 40 Gb/s coherent polarization multiplex QPSK receiver. Although the target transmission rate has not been reached due to underestimated chip interconnection difficulties, we are optimistic to continue our work with more advanced modulation formats (M-QAM) for which hardware-efficient receiver algorithms already have been published[17]. For commercial systems, it is very likely that single-chip solutions with integrated ADCs will avoid the critical chip-to-chip interconnections[7].

References

1 R. Noé, *IEEE PTL*, **17**, 887, (2005)
2 S. Savory, *Proc. ECOC'08*, Mo.3.D.1, (2008)
3 T. Pfau et al., *Proc. OFC'08*, OTuM3, (2008)
4 L. Nelson et al, *Proc. OFC'08*, PDP9, (2008)
5 H. Takahashi et al, *Proc. ECOC'08*, Th3E4 (2008)
6 P. Winzer et al, *Proc.ECOC'08*, Th3E5, (2008)
7 H. Sun et al., *Opt. Express*, **16**, 873 (2008)
8 L. E. Nelson et al., *IEEE JLT*, **27**, 158 (2009)
9 T. Pfau et al., *Opt. Express*, **16**, 866 (2008)
10 S. Hoffmann et al., *IEEE PTL*, **21**, 137 (2009)
11 A. J. Viterbi et al., *IEEE Trans. Inf. Theory,* **29**, 543, (1983)
12 M.-L. Boucheret et al., *ECSC-3*, 155, (1993)
13 S. Hoffmann et al., *Proc. OAA/COTA2006*, CThC6, (2006)
14 O. Adamczyk et al., Electronics Letters, **44**, 895, (2008)
15 V. Herath et al. *Proc. OFC'09*, JThA63, (2009)
16 M. El-Darawy et al. Proc. ECOC 2008, We.3.E.4, (2008)
17 T. Pfau et al. *IEEE JLT*, **27**, 989, (2009)