(article starts on next page)

Dynamic Equalizer Power Dissipation Optimization

Christoffer Fougstedt[1], Pontus Johannisson[2], Lars Svensson[1], and Per Larsson-Edefors[1]

[1]*Dept. of Computer Science and Engineering,* [2]*Photonics Laboratory,*
Chalmers University of Technology, SE-412 96 Göteborg, Sweden
chrfou@chalmers.se

Abstract: We investigate power dissipation/performance trade-off in a 28nm ASIC implementation of a parallel dynamic equalizer. Over 50% dissipation improvement is possible by sample pruning during filter updates, with only minor tracking-performance reduction.
OCIS codes: (060.1660) Coherent communications; (060.4510) Optical communications

1. Introduction

Coherent fiber-optic communication receivers rely on computationally intensive digital signal processing (DSP) algorithms to compensate for signal impairments. The compensation of dynamic link impairments such as polarization drifts is, together with chromatic dispersion (CD) compensation and forward error correction, a major cause of power dissipation in the receiver ASIC [1, 2]. Previously published work includes a review of the power dissipation of a commercial receiver ASIC [1], a power dissipation estimate based on operation complexity [2], and an investigation of the possible power dissipation reduction from a sign-based update algorithm in the context of few-tap low-resolution dynamic equalizers [3]. In contrast to these previously published papers, we study trade-offs between performance and power dissipation with emphasis on the filter tap update mechanism in equalizers with high symbol rates.

It is well known that mechanical perturbations can cause rapid polarization transients in optical fibers. Dispersion-compensating fibers (DCFs) can be particularly sensitive in this respect due to the spooling of the fiber [4]. DSP-based CD compensation avoids the use of DCFs; designing the dynamic equalizer according to the polarization tracking demands presented by DCFs may lead to an unnecessarily high power dissipation.

In this work, we evaluate the performance of a fully parallel 28-GBd dynamic equalizer, in which *sample pruning* is employed, i.e., the number of samples used for the FIR filter updating is reduced. A MATLAB/VHDL co-simulation approach is used, where an equalizer implemented in VHDL is used in a MATLAB-based coherent fiber-optic system model. We perform synthesis and power dissipation simulations using a 28nm process, present detailed power estimates for the dynamic equalizer, and show that the power dissipation can be significantly reduced with only a minor impact on the equalizer's ability to track fast polarization rotations.

2. Dynamic Equalizer Structure and Subsystems

Dynamic equalizers are often implemented as an FIR filter bank, where the filter taps are continuously updated using an optimality criterion. The filter input is a block of polarization-multiplexed (PM) samples $\mathbf{x} = (\mathbf{x}_1^{\mathrm{T}}, \mathbf{x}_2^{\mathrm{T}})^{\mathrm{T}}$, where the vector $\mathbf{x_1}$ ($\mathbf{x_2}$) contains x-polarized (y-polarized) samples. The PM output sample, $\mathbf{y} = (y_1, y_2)^{\mathrm{T}}$, is obtained using the filter tap coefficients, $\mathbf{h}_{ij}, i, j \in \{1, 2\}$, according to $y_1 = \mathbf{h}_{11}^{\mathrm{T}}\mathbf{x}_1 + \mathbf{h}_{12}^{\mathrm{T}}\mathbf{x}_2$ and $y_2 = \mathbf{h}_{21}^{\mathrm{T}}\mathbf{x}_1 + \mathbf{h}_{22}^{\mathrm{T}}\mathbf{x}_2$, respectively [5, Eq. (6)]. This constitutes the *FIR filtering* subsystem of the dynamic equalizer.

The FIR filter taps are typically updated using the stochastic gradient descent method, which requires a gradient estimate. For the constant modulus algorithm (CMA), the cost function [5, Eq. (2)] must be differentiated with respect to the tap coefficients [5, Section 3.2]. Using the chain rule, this differentiation consists of two steps: *error calculation* by differentiating the cost function with respect to the output signal, and *gradient calculation* by multiplying the error with the input vectors. The resulting tap update rules are $\mathbf{h}_{ij}^{(k+1)} = \mathbf{h}_{ij}^{(k)} - \mu e_i \mathbf{x}_j^*$, where μ is the step size, e_i is the error, and k is the iteration number [5, Eqs. (7)–(10)]. The multiplicative complexity is the same for both FIR filtering and gradient calculation if the step size multiplication can be implemented as a right shift.

It should be noted that the only algorithm-specific subsystem is the error calculation, which is based on the CMA cost function. We will show that the error calculation represents only a small fraction of the total power dissipation, which implies that the majority of the conclusions of this paper are not CMA specific.

An ASIC implementation requires parallelization to meet the throughput requirements. Thus, a gradient estimation is performed for calculated output samples and averaged over each block, and the tap coefficients are updated once per parallel block with a delay chosen based on implementation considerations; this in contrast to the sample-by-sample algorithm where the tap coefficients are updated before the next output sample calculation.
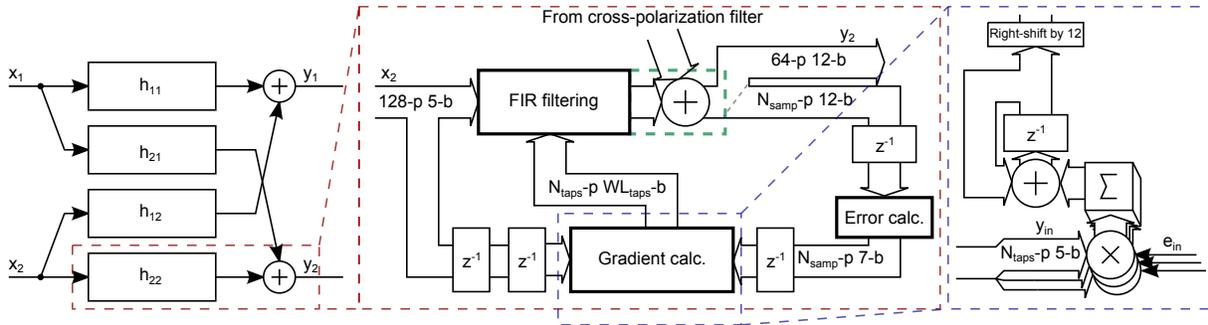
Fig. 1: Block diagram of the parallel implementation of the dynamic equalizer. Registers are represented by z^{-1}. Cross-polarization addition is included in FIR filtering block, as indicated by dashed green line around the adder.

3. VHDL Implementation

A block diagram of the equalizer implementation is shown in Fig. 1, in which internal word-lengths and parallelization factors are annotated with the suffixes -b and -p respectively. N_{taps} and WL_{taps} refer to the number of taps used and the tap resolution, respectively, and N_{samp} refers to the number of samples used for gradient estimation. The internal signals are right-shifted at each explicit word-length reduction. Increases in word-length for preventing overflow in arithmetic units are not shown in the block diagram.

The implemented algorithm updates the FIR filter coefficients once per processed parallel block. The coefficient update is delayed to allow for pipelining, as a non-pipelined filtering and update chain would limit the attainable clock rate of the system. Pipelining also reduces power dissipation; switching-activity increasing glitches, introduced by the arithmetic units, are stopped by clocked registers instead of being propagated to the input of other blocks. The delay register placements are shown in Fig. 1. In addition to dynamic channel compensation, the dynamic equalizer performs matched filtering and a reduction of the sampling rate; the oversampled input signal is downsampled from two to one sample per symbol. Downsampling by a factor of two allows for a reduction in hardware complexity in the FIR filter and update algorithm by a factor of two, as only calculations for the needed outputs are required.

Reducing the number of samples used in the update algorithm lowers the circuit complexity at the expense of giving a noisier gradient estimation, decreasing tracking performance. Each vector of N_{taps} input samples is multiplied by the corresponding error to provide a gradient estimate. The calculated gradients are summed using an adder tree to provide a scaled average; using fewer samples for each error and gradient calculation reduces the effective step size. The averaged gradient estimation is added to the least significant bits of the coefficient accumulators. The output of each accumulator is right-shifted 12 bit positions to adjust step size, truncated to the coefficient word-length, and fed into the FIR filter coefficient inputs. The magnification of the gradient calculation block in Fig. 1 shows the principle of operation. Registers for handling block-edge merging are not shown.

4. Results and Discussion

The VHDL dynamic equalizer implementation is based on a parameterized approach which allows us to create different equalizer configurations that vary in terms of word-lengths, parallelization, and number of samples used for filter updates. 128-way parallelization is assumed throughout the paper, implying that each block of 128 samples is filtered and downsampled to a block of 64 symbol-spaced samples. The target throughput of 28 GBd is achieved using a clock rate of 437.5 MHz. The different equalizer configurations are first verified as register-transfer level (RTL) descriptions in Mentor Graphics Questa simulator, then synthesized to netlists in Cadence RTL Compiler, using a 28nm 1.0-V fully-depleted silicon-on-insulator standard-cell library (regular V_T, typical process corner, and 25 °C). The physical layout estimation feature of RTL Compiler is used to provide more accurate results, and carry-save optimization is used to optimize the multiply-accumulate units and adder trees in the implemented circuits.

Internal word-length requirements are investigated using a MATLAB-implemented block-processing algorithm without delayed update, and the determined word-lengths are used in the VHDL implementations. The implemented VHDL equalizer is then simulated to verify the absence of internal overflow. Internal gradient and error calculation word-lengths are set to be short without increasing the power penalty beyond 0.35 and 0.25 dB for 6- and 7-bit filter tap word-lengths, respectively, in a 16-tap filter. The same word-length configuration is used in the 8-tap filter.
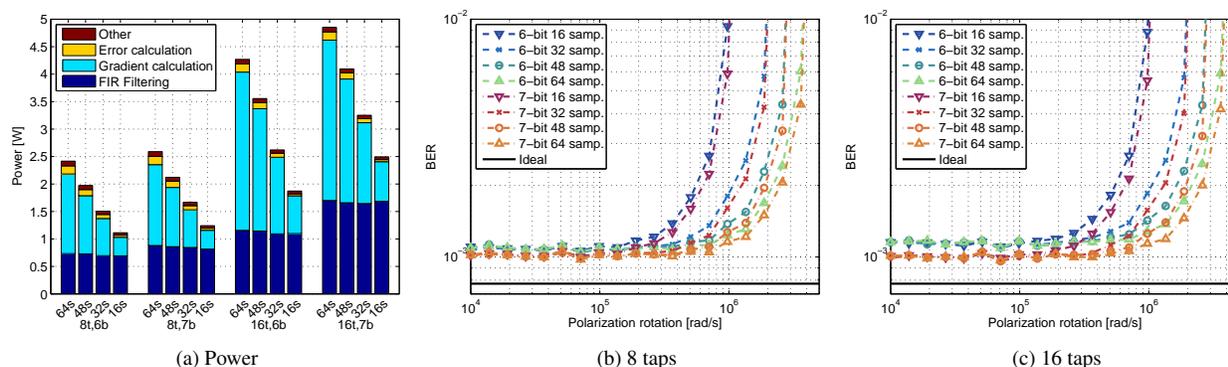
Fig. 2: Power dissipation of the synthesized equalizer configurations (Fig. 2a), and BER vs polarization rotation speed (Fig. 2b, 2c). *Other* represents power dissipation in circuitry not included in the three specific subsystems (Sec. 2).

Performance is evaluated using a MATLAB-based coherent fiber-optic system model in which the VHDL dynamic equalizer implementation is integrated. The system uses QPSK modulation at 28 GBd with 2 samples per symbol, and root-raised-cosine pulse-shaping with a roll-off factor of 0.25. An AWGN channel is used with an added deterministic polarization rotation around the S_3-axis on the Poincaré sphere, where the rotational speed is used to quantify the equalizer tracking performance. The MATLAB system model outputs the fixed-point equalizer inputs and calls the Questa simulator which simulates the VHDL equalizer. The equalizer output is then read by the MATLAB system model. This setup allows parameter studies and BER calculations. Each data point consists of BER calculations from a 2^{21}-bit simulation; the beginning of the vector is truncated to discard the equalizer convergence time.

Power dissipation is analyzed by simulating the equalizer netlists with input data generated in the system model and dumping internal circuit switching activity statistics for each subsystem (FIR filtering, error calculation, and gradient calculation). The switching activity is then back-annotated to the netlist and power estimation is performed in Cadence RTL Compiler. The resulting power dissipation for the implemented filters is shown in Fig. 2a, where N_{taps}, WL_{taps}, and N_{samp} are listed using the suffixes t, b, and s, respectively. Clearly, reducing complexity in the update algorithm gives significant reduction in power dissipation (above or close to 50% overall). The gradient calculation dominates the power dissipation when all samples are used, even though the multiplicative complexity is the same as for FIR filtering. This is due to the signal statistics; the inputs to the filtering block switch rapidly, while the tap inputs are relatively static and change slowly, but both the input and the error signal switch rapidly in the tap update algorithm. This shows that multiplicative complexity provides a poor estimation of power dissipation. BER as a function of polarization rotation speed at $E_b/N_0 = 7$ dB is shown in Fig. 2b and Fig. 2c for 8-tap and 16-tap equalizers, respectively.

All filter configurations can track polarization rotations of at least 100 krad/s without increase in BER. The BER impact from tap word-length is higher in the 16-tap configurations than in the 8-tap configurations; quantization errors affect each tap and a higher number of taps will thus introduce larger errors in the output. We note that accurate CD compensation is important from a power dissipation perspective, as the dynamic equalizer otherwise needs to be longer to be able to handle residual CD.

5. Conclusion

Detailed power dissipation figures of a real-time capable dynamic equalizer VHDL implementation have been presented, showing that multiplicative complexity is a poor estimate of power dissipation. We have shown that a reduction of the number of samples used in the FIR filter tap updating can reduce power dissipation by more than 50%, depending on the amount of sample pruning, while still maintaining tracking performance above 100 krad/s.

References

[1] D. Crivelli *et al.*, "Architecture of a single-chip 50 Gb/s DP-QPSK/BPSK transceiver with electronic dispersion compensation for coherent optical channels," IEEE Trans. Circuits Syst. I, Reg. Papers **61** (2014).

[2] B. Pillai *et al.*, "End-to-end energy modeling and analysis of long-haul coherent transmission systems," J. Lightw. Technol. **32** (2014).

[3] D. Cardenas *et al.*, "Reducing the power consumption of the CMA equalizer update for a digital coherent receiver," in "Opt. Fiber Commun. Conf. (OFC)," (2014), p. Th4D.5.

[4] G. Soliman *et al.*, "Measurement and simulation of polarization transients in dispersion compensation modules," J. Opt. Soc. Am. A **27**, 2532–2541 (2010).

[5] P. Johannisson *et al.*, "Modified constant modulus algorithm for polarization-switched QPSK," Opt. Express **19**, 7734–7741 (2011).