International Association of Scientific Innovation and Research (IASIR)
(An Association Unifying the Sciences, Engineering, and Applied Research)

ISSN (Print): 2279-0020
ISSN (Online): 2279-0039

International Journal of Engineering, Business and Enterprise Applications (IJEBEA)

www.iasir.net

Implementation of Coherent Optical Digital Communication Systems Using Digital Signal Processor & FPGA

Miss. Preeti V. Murkute[1], Mr. A.H.Karode[2]
[1]ME 1st year Digital Electronic, [2]Assistant Professor,
Electronics and Telecommunication department
S.S.B.T's C.O.E.T. Bambhori,
North Maharashtra University
Jalgaon, Maharashtra, India

Abstract: *The emergence of capable semiconductor processes has allowed digital signal processing to extend the application range of high-capacity optical systems. Coherent communication systems have dominated the world of wireless communication almost since its beginnings. The practical optical coherent communication systems became feasible only recently. Digital signal-processing-based coherent optical communication systems are widely viewed as the most promising next generation long-haul transport systems. One of the biggest challenges in building these systems is the implementation of signal processors that are able to deal with signalling rates of a few tens of giga-samples per second. This paper covers implementation options and design considerations with respect to hardware realization and DSP implementation.*

Keywords*: Digital signal processors, coherent communication, quadrature phase-shift keying, FPGA, QPSK*

I. Introduction

We have been studying about the coherent systems, communication systems, digital processing and implementation of these techniques. Coherent detection offers the advantage of access to the amplitude and phase of the optical electric field in the electronic domain at the receiver. This allows digital linear filters to compensate the linear channel transfer functions. Today's motivation to revive coherent concepts in optical communication is twofold. First, coherent receivers enable reliable data transmission with much higher spectral efficiency than conventional direct-detection systems, and second, coherent receivers can compensate for linear impairments, most notably, polarization-mode dispersion (PMD) to a degree that is out of reach for conventional systems. Also, the technical difficulties that the first generation of coherent systems in optical communications faced have been lessened. This is caused by two developments.

First of all, the symbol rate to carrier frequency ratio of modern optical communication systems approaches the ratio that is commonly used in wireless systems. For a system that transmits at data rate of 100 Gb/s in two polarization orientations utilizing QPSK (Quadrature Phase Shift Keying) signaling, the symbol rate is 25 GBd. With a carrier frequency of roughly 200 THz, the symbol rate to carrier frequency ratio is 1.25e-3. This indicates that it is possible for optical systems to achieve similar phase noise to symbol rate ratios, as in wireless systems.

Second, the performance of digital signal processing (DSP) equipment has been improved dramatically over the past two decades, which makes it feasible to implement the complex signal processing steps required to synchronize to the received signal in digital domain. Implementations of optical coherent receivers have been demonstrated in CMOS-based application specific ICs (ASICs).

Albeit a coherent optical communication system can utilize single or multiple carrier [e.g., orthogonal frequency-division multiplexing (OFDM)] transmitter and any modulation format, with QPSK being the most popular and higher order quadrature amplitude modulation (QAM) and phase-shift keying (PSK) systems under investigation, this paper will concentrate on single-carrier frequency-domain-equalized systems, which has become more popular in the wireless domain as well. The modulation format discussed here will be QPSK. Phase coherence between a data signal and the reference is typically established at the receiver side.

II. Implementation Considerations

Today's FPGAs offer processing speeds in the order of a few 100 MHz. The achievable processing speed for an ASIC using the same CMOS generation is typically higher by a factor of about two to three.

Nevertheless, the processing speeds available in today's technologies are about two to three orders of magnitude smaller than the data rates in optical communication systems. The maximum achievable processing speed of a digital circuit is given by the longest time a signal needs to travel between two clocked storage elements (e.g., flip–flops). The path between these two storage elements is called the "critical path."

There are two commonly used techniques to reduce the length of the critical path, and therefore, increase processing: pipelining and parallel processing.

Pipelining reduces the critical path by inserting additional retiming elements along the signal path in a manner that does not alter the result of the processing but of additional latency. This only allows an increase of processing speed up to the maximum speed of a single element (gate in an ASIC or lookup table (LUT) for an FPGA). For being able to process data at multiple gigabit per second up to 100 Gb/s and beyond, a parallel processing structure has to be implemented. Unfortunately, not all algorithms can be parallelized without modifications and loss of performance. In general, all structures that can be pipelined can also be processed in parallel. Algorithms that are time invariant can simply be parallelized without loss of performance by instantiating the circuitry that implements the algorithm multiple times. Often, the complexity can be reduced by sharing resources between multiple instances.

An example of a structure that can easily be parallelized is a finite-impulse response (FIR) filter with constant coefficients. If the filter coefficients are not constant, e.g., within an adaptive filter structure, there might not be an equivalent parallel structure, e.g., when the update of the filter coefficients is performed might not be an equivalent parallel structure, e.g., when the update of the filter coefficients is performed once per sampling period. If one is willing to compromise on update speed by accepting an update rate once every clock cycle, with the clock cycle being 1/n times the sampling period, with n the parallelization factor, an equivalent structure can be implemented.

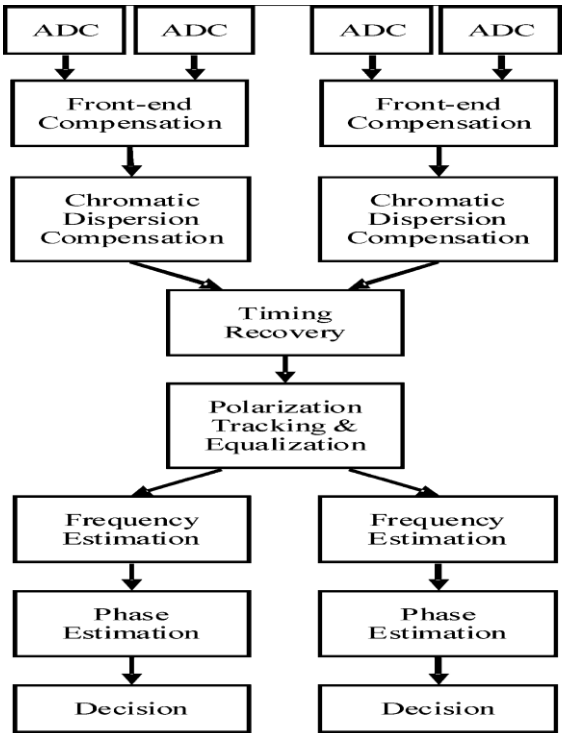

Figure1: Digital signal processing steps

III. Quadrature-Imbalance Compensation

In an initial step, impairments of the optical and electrical frontend are compensated. One example is the correction of quadrature imbalance stemming from imperfect phase control in the optical hybrid. Quadrature-imbalance compensation is well known in wireless communications and has been proposed to be used in optical communications as well. If there is no amplitude and offset error of the in-phase and quadrature component of the signal, quadrature-imbalance compensation can be performed by first measuring the cross correlation between the in-phase (*I*) and quadrature (*Q*) components of the received signal, which is proportional to the sine of the phase error ψ of the optical hybrid

$$\langle \boldsymbol{I(t)Q(t)} \rangle = \frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{sin(\psi)} \qquad (\mathbf{1})$$

where *I* and *Q* are assumed to be normalized. The components can then be transformed in corrected orthogonal components *I* and

$$\begin{pmatrix} \boldsymbol{I(t)} \\ \boldsymbol{Q(t)} \end{pmatrix} = \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \boldsymbol{tan\psi} & \boldsymbol{sec\psi} \end{pmatrix} \begin{pmatrix} \boldsymbol{I'(t)} \\ \boldsymbol{Q'(t)} \end{pmatrix} \qquad (\mathbf{2})$$

Equations (1) and (2) can be implemented in a feed-forward structure. This has two major drawbacks. First, the trigonometric functions have to be implemented in an LUT, and second, the normalization of the two components has to be performed accurately; as any error in the normalization will lead to quadrature imbalance.

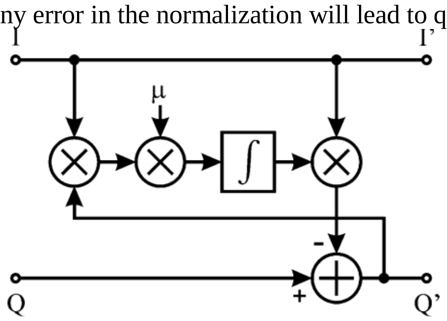

Figure 2: Circuit for quadrature-imbalance compensation

The circuit shown avoids these drawbacks by employing a feedback structure. The cross correlation is measured after the actual compensation, weighted with a convergence factor μ and integrated. After convergence is achieved, the cross correlation is zero in average and the output of the integrator is constant and, according to (1), it is proportional to the sine of the phase error. This result is then multiplied with the *I*-tributary and added to the *Q*-tributary to yield the corrected output. Note that, according to (2), the result should also be divided by the cosine of the phase error to yield the correct amplitude. This step is omitted in Fig. 2, as it would require an LUT and another multiplier. If a gain control circuit is placed after the quadrature-imbalance compensation, this would be automatically compensated for.

IV. CD Compensation

It is advisable to split the equalization of the received signal in two steps. First, perform a static or slowly adaptive equalization on each polarization tributary separately, and second, perform a fast adaptive joint equalization on both polarization tributaries.

The first equalizer is typically chosen to have a much longer impulse response and can be used to compensate for quasistatic effects as CD or frequency response of the optical frontend. The second one having a shorter impulse response but a faster adaptation speed is typically used for polarization tracking, equalization of PMD, as well as residual CD not compensated for by the static equalizer.

Typically, equalizers for data rates considered here employ digital block filters. Block filtering involves the calculation of a finite set, or block of output values based on a finite set of input values. This can be performed in time domain or equivalently in frequency domain [11]. Algorithms have been developed for block filtering to achieve identical outputs as sequential filtering, most notably, the “overlap-and-save” method and the closely related “overlap-and-add” method.

Let us assume that the input data sequence is partitioned in blocks of length *n* and that *k* is the impulse response length of the desired filter function. In case of overlap-and-save, *n* input samples are concatenated with *k* symbols from the next block, and then, convoluted with the impulse response. The first *k* samples of the output of the convolution are not used, while the remaining *n* samples constitute the correct filter output.

Therefore, this method is often also referred to as “overlap-and dump” or “overlap-and-scrap.”

In case of overlap-and-add, *n* input samples are padded with *k* zeros before being convoluted with the impulse response of the desired filter. After convolution, *k* trailing samples are stored to be added to the *k* leading samples of the following result of the convolution. Overlap-and-add is typically slightly more efficient in terms of implementation complexity and is often chosen when the filter response is static or changes only slowly with time. For fast adaptive filters, the overlap-and-save method is preferred, because for the overlap-and-save method, each output block is the filtering result of exactly one impulse response function, while in case of the overlap-and-add method, the portion that is saved from the previous result to be added to the current result might have been calculated with a different impulse response function if the filter function changed in-between two clock cycle.

Block filtering can be efficiently implemented in frequency domain, especially if the impulse response length is comparable to the block length.Frequency-domain filtering requires the implementation of discrete Fourier and discrete-inverse Fourier transforms. The most commonly used algorithm to implement Fourier transforms in hardware is the Cooley–Tukey fast Fourier transform (FFT) algorithm [4].Basic idea of the Cooley–Tukey algorithm is to break up a transformation of length *N* in two transformations, each of length *N*/2 (Danielson–Lanczos Lemma). This can be done recursively, until one reaches a transform of trivial size (two, four, or eight, for instance). It is possible not only to divide up the FFT in two parts, as described earlier (radix-2).

Very common are also radix-4 implementations, where in each step, the FFT is split in four sub-FFTs, or mixture of radix-2 and radix-4 (split radix), which are most suitable for hardware implementation.

V. Timing Recovery

The received data and the sample timing need to be synchronized so that a fixed ratio (typically two samples per symbol) is established. Timing recovery comprises two components, a timing-error detector and an interpolator. Interpolation can easily be implemented utilizing an FIR filter.

One of the most commonly used timing-error detectors in digital communication is the Gardner timing-error detector. Unfortunately, if PMD causes differential group delays approaching half symbol duration, the Gardner timing-error detector will fail. Extensions of the Gardner timing-error detector have been proposed to overcome this limitation.

VI. Polarization Tracking

Polarization tracking and PMD equalization is typically performed using a two-in two-out adaptive filter. An adaptive filter can be partitioned in three parts: the actual filter bank, an error estimator, and a device for updating the filter coefficients. The filter itself has typically a rather short impulse response. Because it needs to follow arbitrary polarization rotation, a rather fast update of the coefficients is required. Therefore, as discussed earlier, an overlap-and-save implementation is preferable.

In a second step, the error of the signal coming from the filter bank needs to be estimated. There are a number of techniques available for error estimation, namely insertion of training symbols, decision feedback, or measuring a known property of the signal. The former have good tracking properties but require the inclusion of carrier synchronization in the feedback loop. The latter one has advantages with respect to loop delay, and therefore, potentially offers faster tracking speed. A very popular measure is the constant modulus criterion. The constant modulus criterion penalizes deviation of the amplitude of the equalized signal from a desired fixed value.

It is obvious that this criterion is optimally suited for PSK-modulated signals. Actually, this criterion can also be utilized in QAM-modulated systems, albeit with a penalty with respect to noise and convergence speed.

In a third step, from the estimated error, updated filter coefficients have to be calculated. Several algorithms for this are known in literature, for instance, the Wiener–Hoff solution, the method of steepest descent. Most practical from an implementation standpoint of view is the LMS algorithm. The idea of the LMS algorithm is to estimate the gradient of the error by partial derivatives of the mean-squared error with respect to the filter coefficients. The gradient estimates are calculated from instantaneous measures of the error, i.e., the difference between the desired amplitude and the instantaneous signal amplitude after the adaptive filter.

In each step, the filter coefficients are updated by adding a small portion proportional to the negative gradient estimate. A weighting factor μ is again utilized for controlling the adaptation speed and residual error of the adapted filter coefficients. Exact formulation of this algorithm and comparison with a decision-feedback structure can.

VII. Carrier Synchronization

Carrier synchronization (i.e., frequency offset and phase-error estimation and correction) is probably the most comprehensively treated topic in literature, as this is the minimum processing that needs to be performed in any DSP-based coherent receiver. The first published work therefore concentrated on this topic. Frequency offset correction and phase-error correction is conceptually very similar, as both involve the estimation of an error, filtering of that estimate, and correction of the data utilizing the filtered estimate.

Phase synchronization typically comprises two steps.

First, removal of the modulated information to obtain an instantaneous phase estimate, and second, filtering of the phase estimate to minimize the influence of noise.

In most communication systems, information removal is performed by employing a decision-directed scheme, where the difference between a symbol before and after decision is taken as instantaneous estimate for the phase error. Decision-directed schemes typically utilize feedback, which poses a challenge for pipelining and block processing. In a direct parallelization of the decision-directed feedback structure, the feedback delay is multiplied by the parallelization factor, which leads to an equivalent reduction in phase noise tolerance. Look-ahead techniques can be utilized to improve performance.

In case of PSK-modulated signaling, information removal can also be performed by applying power-law nonlinearity. This is a feed-forward technique that can be easily implemented in a block-processing scheme. The same hold for the post-estimation filtering, as a simple FIR filter can be utilized.

The most important problem encountered in broadcasting via terrestrial transmitters is the interference from other broadcasters. In principle, each broadcaster has a different radio frequency (planned by the public authority) in a common reception area to avoid interference from each other. However still there are two problems: spurious radiation of adjacent channels and fringe reception.

Fringe reception is unintended reception under certain weather conditions. The exceptionally long-range reception means that the receiver may be tuned to more than one transmitter (transmitting at same frequency) at the same time. These transmitters may transmit programs of different broadcasters as well as the programs of the same broadcaster. In analogue transmission, even the transmitters transmitting the very same program interfere each other because of phase differences of the incoming signal, but in digital transmission the transmitters transmitting the same program in the same channel may reinforce each other.

VIII. FPGA Implementation of Signal Processing Algorithms in Coherent Optical Systems [5]

Coherent optical communications carry several advantages over intensity modulated direct detection systems, namely the ability to use phase modulated (M-QPSK) and multi-level constellations (M-QAM), due to the preservation of the electric field from the optical domain to the electrical domain, provided the sampling is at least at Nyquist rate. Additionally, it enables quasi-exact compensation of linear transmission impairments, such as chromatic dispersion (CD) and polarization mode dispersion (PMD) by a linear filter, which can operate adaptively to overcome time-varying impairments.

These systems have gained renewed interest due to the availability of high speed digital signal processing, which allows for complex operations to be carried out in the digital domain, enabling high potential for reconfigurable software defined optical receiver. AD converters will be able to satisfy the required high sampling rates in the near future for optical long-haul high speed transmission systems. Moreover, as soon as sufficiently high speed data converters are available, field programmable gate arrays (FPGA) are a very flexible implementation platform.

Considering that the Local Oscillator (LO) phase needs to be locked to the signal phase, to avoid the difficulties associated with the Optical Phase Locked Loop (OPLL), the synchronization can be done in the DSP, by digital phase estimation techniques, allowing for a free running LO. Algorithms suitable for phase estimation and dispersion compensation have been studied.

IX. Adaptive Equalization

A. Constant Modulus Algorithm – CMA

The CMA is the most used algorithm for adaptive equalizers, essentially because of its robustness and ability to converge prior to phase recovery. The signal at the equalizer output is obtained through convolution of the equalizer coefficients with the digitized signal.

$$y(n) = w^H(n).u(n) \qquad [1]$$

Then the error signal is calculated as follows:

$$e(n) = y(n).(R_2 \quad |y(n)\,|^2) \qquad [2]$$

Where R2 is a constant dependent on the selected constellation.

The coefficient update is then given by:

$$w(n+1) = w(n) + \mu.u(n)e^*(n) \quad [3]$$

where µ is the algorithm step size. It is important to consider the initialization of the coefficients for successful convergence. A centre spike initialization is required, where the central coefficient is equal to the unity, while all other coefficients are set to zero. This algorithm provides good performance, but disregards the signal phase, causing the constellation to twist. On the other hand, the next algorithm takes the phase into account.

B. Least Mean Squares - LMS

The LMS is very similar to the CMA, except for the signal error calculation, where an extra module is used to calculate symbol decisions at the equalizer output. The error can be calculated as:

$$e(n) = d(n) - y(n) \qquad [4]$$

where d (n) is the decided symbol.

The LMS initialization might be done in two ways. A training sequence is used for initial convergence, the LMS being switched to Decision Directed (DD) mode thereafter. However as training sequences are difficult to provide, a more elegant approach can be used, where the LMS coefficients are initialized to the coefficients obtained after CMA convergence.

X. Conclusion and Future Scope

The implementation of adaptive equalization algorithms was carried out in a FPGA platform. We conclude these algorithms might be implemented with good performance given that a sufficient number of bits is used. We have successfully initialized the LMS algorithm with the coefficients resulting from CMA convergence. Furthermore, for high fiber distances an equalization module with fixed coefficients would be required.

References

1. "Real-Time Implementation of Digital Signal Processing for Coherent Optical Digital Communication Systems", Andreas Leven, Senior Member, IEEE, Noriaki Kaneda, Senior Member, IEEE, and Stephen Corteselli, ieee journal of selected topics in quantum electronics, vol. 16, no. 5, september/october 2010
2. Book title- "SIGNALS AND SYSTEMS", Edition 2, by Alan V. OPPENHEIM, Alan S. Willsky Massachusetts Institute Technology with S. Hamid Nawab, Boston University, Prentice-Hall International,Inc
3. "Digital Signal Processing", A Practical Guide for Engineers and Scientists by Steven W. Smith

4. “Digital Signal Processing Principles, Algorithms and Applications”, Third Edition by John G. Proakis Northeastern University, Dimitris G. Melonakos ,Boilon College, Prentice-Hall International,Inc
5. “FPGA Implementation of Signal Processing Algorithms in Coherent Optical Systems”, N. M. Pinto, L. M. Pessoa, J. C. Ferreira and H. M. Salgado
6. “Digital Signal Processing for Optical Coherent Communication Systems”, by Xu Zhang Delivery Date: April 27th 2012 ,DTU Fotonik, Department of Photonics Engineering, Technical University of Denmark, Building 343 2800 Kgs. Lyngby,DENMARK
7. JOURNAL OF LIGHTWAVE TECHNOLOGY, VOL. 26, NO. 10, MAY 15, 2008 1309,Phase Noise Effects on High Spectral Efficiency, Coherent Optical OFDM Transmission ,Xingwen Yi, William Shieh, Member, IEEE, and Yiran Ma
8. “Coherent Receivers for Practical Optical Communication Systems”, Andreas Leven, Noriaki Kaneda1, Ut-Va Koc, Young-Kai Chen Bell Laboratories, Lucent Technologies, 600 Mountain Ave., Murray Hill, NJ 07974, aleven@lucent.com 1: Optical Networking Group, Lucent Technologies, 101 Crawfords Corner Rd., Holmdel NJ 07733
9. “SM320C6455-EP”,FIXED-POINT DIGITAL SIGNAL PROCESSOR
10. Manual, Literature Number: SPRS462B,SEPTEMBER 2007–Revised JANUARY 2008,PRODUCTION DATA information is current as of publication date, Products conform to specifications per the terms of the Texas Instruments standard warranty.
11. “Optical OFDM Basics” , Qi Yang, Abdullah Al Amin, and William Shieh IEEE.
12. Coherent Digital Polarization Diversity Receiver for Real-Time Polarization-Multiplexed QPSK Transmission at 2.8 Gb/s Timo Pfau, Ralf Peveling, Jérôme Hauden, Nicolas Grossard, Henri Porte, Member, IEEE, Yaakov Achiam, Sebastian Hoffmann, Selwan K. Ibrahim, Member, IEEE, Olaf Adamczyk, Suhas Bhandare, Member, IEEE, David Sandel, Mario Porrmann, and Reinhold Noé, Member, IEEE .

XI. Acknowledgments

I am very thankful to Mr. Atul Karode Sir who has been guiding me throughout my research paper work. My parents and friends for their support and coordination.