

Document downloaded from:

<http://hdl.handle.net/10251/199024>

This paper must be cited as:

Valls Coquillat, J.; Torres Carot, V.; Pérez Pascual, MA.; Almenar Terre, V. (2023).  
Hardware Architecture of a QAM Receiver for Short-Range Optical Communications.  
Journal of Lightwave Technology. 41(2):451-461. <https://doi.org/10.1109/JLT.2022.3217357>



The final publication is available at

<https://doi.org/10.1109/JLT.2022.3217357>

Copyright Institute of Electrical and Electronics Engineers

Additional Information

# Hardware Architecture of a QAM Receiver for Short-Range Optical Communications

Javier Valls, Vicente Torres, Asunción Pérez-Pascual, Vicenç Almenar

## Abstract

Short-reach optical fiber communications systems aim to achieve high throughput, in the order of tens of Gbps. The implementation of these high-speed systems requires parallel processing, which makes low-complexity designs of their subsystems a key to the successful large-scale deployment of this technology. Half-Cycle Nyquist Subcarrier Modulation (HC-SCM) was originally suggested for these systems with the goal of using as much bandwidth as possible and, therefore, achieving high communication rates. Recently, Oversampled Subcarrier Modulation (OVS-SCM) was proposed as an alternative more computationally efficient than HC-SCM and also with a better spectral efficiency. This paper proposes a hardware-efficient architecture for an OVS-SCM receiver, which takes into account the inherent parallel processing of these systems. This receiver takes 16 samples in parallel from a 5 GSa/s analog-to-digital converter with a 3.2 GHz 3 dB bandwidth. Design solutions for the frame detection block, the mixer, the resampler, the fractional interpolator, the matched filter and the timing estimator are presented. Our results show that, compared to the HC-SCM receiver, this proposal reduces the computational load of the downconverter stages by 90%. FPGA implementation results are given to demonstrate that our proposal can be implemented in state-of-the-art devices.

## Index Terms

short-range optical links, FPGA, hardware architecture, receiver

## I. INTRODUCTION

Short-reach optical fiber systems with intensity modulated direct-detection (IM/DD) transmission scheme over standard single-mode fiber (SSMF) is a potential solution to support the continuous increase in data traffic of telecommunications networks. There are three possible solutions to increase the bit rate in any communications system: to make use of more bandwidth, to increase the spectral efficiency by means of modulation formats with more bits per symbol, and to multiplex several data signals. In recent years the move from 10G to 100G in optical links has made use of a mixture of the previously stated solutions. For example, from 10G to 40G multiplexing was the chosen solution by means of four 10G links, whether transmitted using four different wavelengths, or using four different fibers. The road to 100G has been solved with different solutions, each one using a different combination of

Manuscript received XX, XX; revised XX, XX. Grants RTI2018-101658-B100 and PID2021-126514OB-I00 funded by MCIN/AEI/10.13039/501100011033 and by “ERDF A way of making Europe”, by the European Union.

J. Valls, V. Torres, A. Perez-pascual, and V. Almenar are with the Instituto de Telecomunicaciones y Aplicaciones Multimedia, Universitat Politècnica de València, Valencia 46730, Spain (e-mail: jvalls@upv.es; vtorres@upv.es; asperez@eln.upv.es; valmenar@upv.es).

multiplexing (with multiple lines or with multiple wavelengths), bandwidth rise and higher modulation order (PAM4 instead of NRZ). On the other hand, for long haul optic data links the trend is to increase the bit rate by means of improving the spectra efficiency employing high order modulation formats and coherent detection. To this end, Orthogonal Frequency-Division Multiplexing (OFDM) and Nyquist Subcarrier Modulation (SCM) are the solutions evaluated by researchers both from academia and industry; as an example, recently a product making use of SCM technology has been presented [1]. The ever-increasing data bandwidth demand of actual data traffic has triggered a growing interest in the development of high throughput datacenter interconnections (DCI). The main objective in developing next generation DCI is to reach 100 Gb/s making use of components developed and deployed for 10G [2] to reduce costs thanks to the use of mature 10G low-cost optical solutions. To achieve high data rates with a restricted bandwidth the available strategies are: to improve the spectral efficiency (as in long haul optical links) and to multiplex. Moreover, to keep deployment costs low, optical coherent transmission is not foreseeable in the near future, so optical transmission will make use of IM/DD schemes. Therefore, this improvement in spectral efficiency will require the use of powerful digital signal processing solutions [3] and efficient modulation formats such as OFDM [2], [4], [5], [6] and SCM [7], [8], [9], [10]. This last one is the strategy developed in this paper.

In order to take advantage of state-of-art devices, like FPGAs (Field-programmable gate array), digital-to-analog converters (DACs) and analog-to-digital converters (ADCs), to implement very high throughput systems, the symbol rate should be as close as possible to the sampling rate of the selected DACs and ADCs (*i.e.* as close as possible to the Nyquist frequency). With that goal in mind, subcarrier modulation with one subcarrier centered at a frequency equal to half of the symbol rate, and data waveform shaped with a root raised cosine (RRC) Nyquist pulse with a zero roll-off factor was proposed in [11]. This approach was named Half-Cycle SCM (HC-SCM) and its spectral efficiency was improved in [7] implementing a 38 Gb/s 128-QAM scheme, and in [12] by adding dual polarization in a 112 Gb/s 16-QAM system.

Recently, oversampled subcarrier modulation (OVS-SCM) was proposed [9], introducing a non-integer sampling rate higher than 2 and a raised cosine pulse with roll-off factor higher than zero. The authors evaluated a system with a sampling rate of 2.25 samples per symbol (sps), which was obtained with a rational interpolation/decimation rate of 9/4, and a roll-off of 0.1. Compared to HC-SCM, a lower computational load is achieved thanks to the relaxation of the Nyquist filter requirements: the zero roll-off of HC-SCM needs many more coefficients than the non-zero roll-off of OVS-SCM. Additionally, this last approach avoids the interference caused by neighbour bands due to the non-ideal frequency response of the HC-SCM Nyquist shaping filter, which makes possible the use of higher modulation orders and leads to a higher spectral efficiency. Experimental results show that for a bandwidth of 2.5 GHz, OVS-SCM can reach 17.8 Gb/s with 256-QAM, whereas HC-SCM can reach 15 Gb/s with 64-QAM. Nevertheless, systems that work with bandwidths of several GHz require the use of high sampling rate ADCs, which currently provide several samples in parallel. Moreover, due to their maximum clock frequencies (which are in the order of hundreds of MHz), the converted samples must be processed in parallel in state-of-the-art devices, like FPGA devices. The implementation of OVS-SCM systems is heavily influenced by the fact that they must process samples in parallel. In the case of [9], the computational load of an OVS-SCM transmitter and receiver was analyzed, but their hardware architectures were not presented. Moreover, their selected number of sps, and

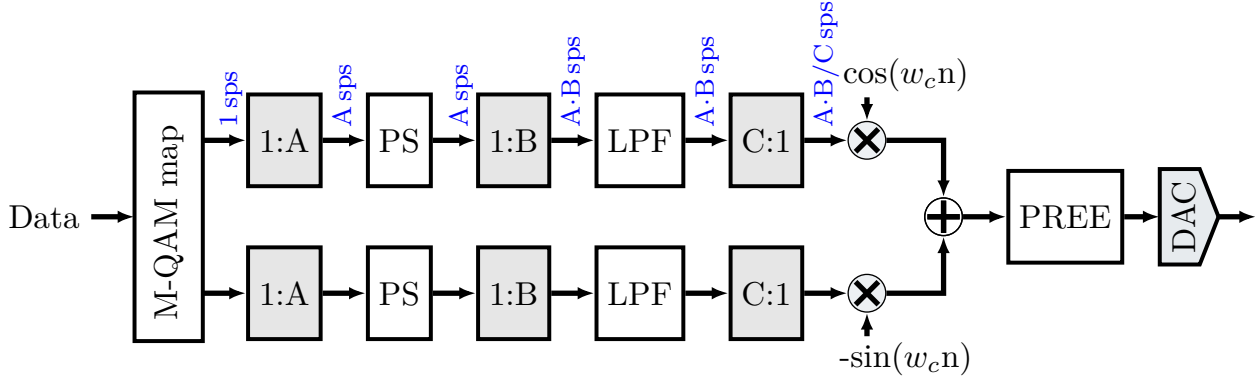


Fig. 1. Conceptual block diagram of the OVS-SCM transmitter. DAC=digital-to-analog converter; sps=samples per symbol; LPF=low pass filter; PS=pulse shaping; PREE=pre-emphasis filter. m:n means a sample rate change where n samples are output for each m input samples.

the associated interpolation/decimation operators do not take into account the parallelism of the processing stages, which increases the complexity and cost of the implementation. Other authors have proposed efficient ways of implementing the transmitter by extensive use of look-up tables [13], [14]. It should be noted that the complexity of the transmitter is much lower than that of the receiver.

In this paper, an efficient hardware architecture for an OVS-SCM receiver is proposed. Here, the rational interpolation/decimation rate is selected taking into account that the ADC delivers 16 samples in parallel, and that by properly selecting this ratio, the implementation of the down-converter stages is simplified. Furthermore, a joint design of the mixer and the fractional rate-changing filter is proposed to simplify the design. Finally, to complete the receiver proposal, the frame and symbol synchronization stages have been included. The frame synchronization is based on a Zadoff-Chu sequence [15] using parallel correlators, which have been modified to reduce their hardware complexity. The symbol synchronization stage is based on [16] and has been adapted to be implemented in parallel.

The rest of the paper has the following structure. In Section II an overview of the OVS-SCM system and the selection of the sample rates for the receiver are introduced. In Section III the architecture of the proposed receiver including its relevant design parameters is described, and the implementation of each block is detailed. In Section IV an analysis of the computational load of the proposal is also performed. In Section V, the experimental setup and the transmission results are presented, and in Section VI, the FPGA implementation results are given. In Section VII a discussion of the results is presented. Finally, in Section VIII we present the conclusions.

## II. OVS-SCM SYSTEM

As commented above, HC-SCM was proposed for short-range fiber-optic communication systems and works with no oversampling, that is, it works with a processing rate of 2 sps. HS-SCM transmitter first generates a base band signal using 2 sps, then it is upconverted and centered at half band. At this point the signal is not oversampled as it fully occupies the entire digital bandwidth. Only if a sinc pulse shaping is employed (which implies a 0 roll-off factor and a brick wall spectrum), aliasing is avoided. On the other hand, the OVS-SCM [9] proposed to increase the

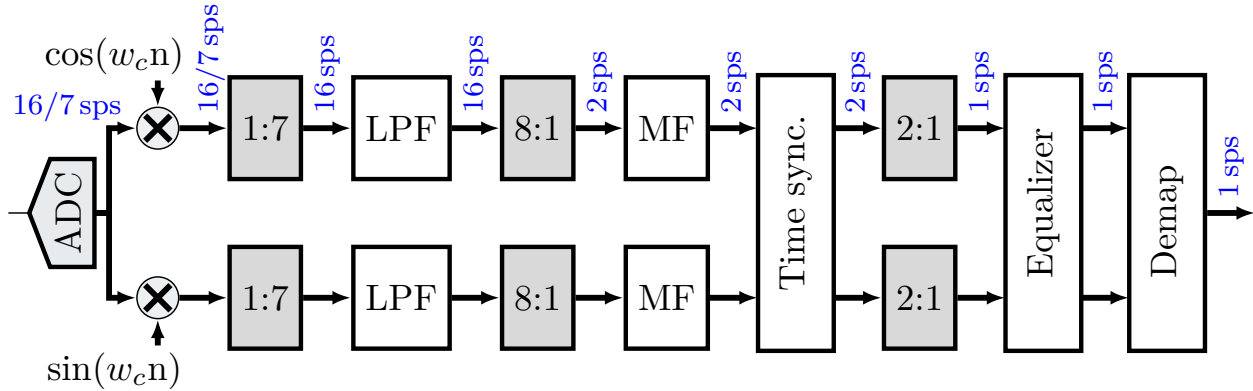


Fig. 2. Conceptual block diagram of the OVS-SCM receiver. ADC=analog-to-digital converter; sps=samples per symbol; LPF=low pass filter; MF=matched filter. m:n means a sample rate change where n samples are output for each m input samples.

sample rate (specifically it worked with  $9/4=2.25$  sps) to simplify the Nyquist filter specifications and, consequently, to reduce the overall complexity of the system. Fig. 1 shows the schematic diagram for the OVS-SCM transmitter. The M-QAM symbols are pulse shaped with a RRC filter working with 3 sps ( $A=3$ ). Then, the pulse shaping filter output was upsampled<sup>1</sup> by 3 ( $B=3$ ) and downsampled by 4 ( $C=4$ ), that is, the final output had a sample rate of  $9/4=2.25$  sps. After that, the samples were mixed with the only carrier with a center frequency of  $f_c=f_s/4$ , and finally, a pre-emphasis was applied in order to compensate for the DAC frequency response.

In this paper, we present the receiver architecture for a system that obtains several samples in parallel from the ADC. The proposal follows the OVS-SCM processing scheme, but, for optimization reasons, different sample rates along the processing chain than those from [9] are employed. Specifically, for the transmitter the pulse shaping stage interpolates by 2 ( $A=2$ ) and then there is a fractional interpolation by  $8/7$  ( $B=8$  and  $C=7$ ), giving a final rate of 2.2857 sps. The selection of these values will be explained in Section III. As a consequence, the receiver processes a  $16/7$ -oversampled QAM signal (see Fig. 2). Following this block diagram it can be seen that after downconversion to baseband, the sample rate is changed to 2 sps by interpolating by 7 and then downsampling by 8. After that, the signal is filtered with a matched filter and its output is time-synchronized by resampling it with a fractional interpolator. Finally, the signal is equalized to compensate for the channel distortion and the demapper block outputs the received bits. The transmission of data in the proposed system is based on frames, whose structure is shown in Fig. 4. Each frame includes a preamble that is used by the receiver for frame and symbol synchronization tasks and for equalization. Each time a frame is received, the timing/symbol synchronization is updated and there is no need to manage underflows/overflows during the reception of a frame, a fact that greatly simplifies the design of a fully digital parallelized receiver.

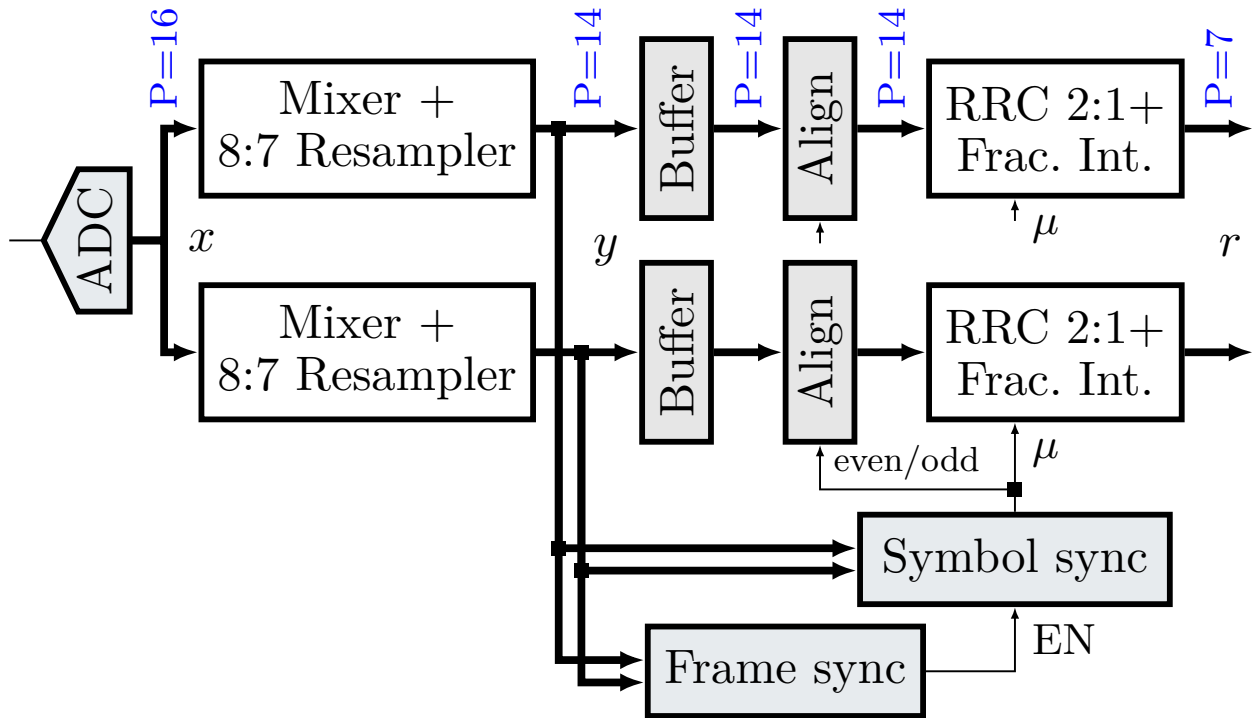


Fig. 3. Implementation diagram of the proposed receiver.  $m:n$  means a sample rate change where  $n$  samples are output for each  $m$  input samples.

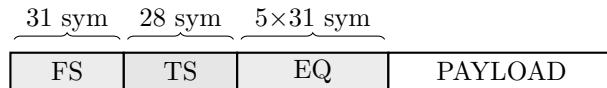


Fig. 4. Data frame distribution. FS: Frame Synchronization; TS: Timing Synchronization; EQ: Equalizer.

### III. RECEIVER ARCHITECTURE

Fig. 3 shows the implementation diagram of the proposed receiver whose blocks are detailed below. The receiver has been optimized for the case of an ADC that delivers 16 samples in parallel. As commented previously, the SCM signal is modulated using a  $f_s/4$  carrier, this fact opens the door to some simplifications in the implementation of the mixer and the resampler stage.

The first change we introduce respect to [9] is to select 2 sps for the RRC matched filter (which corresponds to the pulse shaping filter at the transmitter), and also for the symbol synchronization stage, since this is the lowest sample rate feasible for those blocks. Since the OVS-SCM works with a sample rate higher than 2 sps, a sample rate change is needed to obtain this value at the RRC filter input. For optimization purposes, as the converter delivers a batch of  $P=16$  parallel samples (in Fig. 3,  $P$  is the number of samples that are processed in parallel at any stage), each batch must generate after the resampling process an output of  $N$  samples, where  $N$  must be integer in order to facilitate their parallel computation. Moreover, since the sample rate of those  $N$  samples batch is 2 per symbol,

<sup>1</sup>Along this paper, blocks showing  $m:n$  mean a sample rate change where  $n$  samples are output for each  $m$  input samples

$N$  should be even, so each parallel batch of 16 samples provides an integer number of symbols. It should be noted that the baseband bandwidth is

$$f_{max} = \frac{1}{2T} (1 + \beta), \quad (1)$$

where  $T$  is the symbol period and  $\beta$  is the roll-off factor. With this restrictions in mind,  $N$  can be  $\{14, 12, 10, \dots\}$ , which would give sample rates of  $\{16/7, 16/6, 16/5, \dots\} = \{2.2857, 2.6667, 3.2000, \dots\}$  sps and, according to Eq. 1, maximum roll-off factors of  $\{0.1428, 0.3334, 0.6000, \dots\}$ . We choose  $N=14$  (and roll-off 0.14), since it requires the lowest excess bandwidth. If the bandpass bandwidth of the QAM signal is 2.5 GHz, as is our case, for a roll-off factor of 0.14 Eq. 1 gives a symbol rate of 2.19 Gsym/s.

Next, the different blocks of the receiver are detailed. This proposal includes frame and symbol synchronization blocks, which allow the correct reception of the payload. Those blocks are enabled as soon as a signal is detected in the receiver input. A suitable preamble must precede the payload, so these tasks can be completed. According to the output of the symbol synchronization block, samples can be delayed half symbol (Align block in Fig. 3) before they enter the RRC filter and an appropriate version of the RRC filter coefficients is used in order to correct the timing of the symbols.

#### A. Mixer and resampler filter

As stated before, the received signal is demodulated using a  $f_s/4$  carrier, which leads to a simplified mixer where the required multiplication uses the  $\{1,0,-1,0\}$  and  $\{0,1,0,-1\}$  sequences for the in-phase and the quadrature branches, respectively. Since the ADC provides 16 samples in parallel, they are multiplied by exactly  $16/4=4$  periods of the carrier: as a result, each position of the 16 parallel samples is always “multiplied” by the same carrier value. This fact can be used to further simplify the mixer, which can be implemented as a wired connection, as shown in Fig. 5. It should be noted that the samples marked with an asterisk should be negated, but this negation is implemented later, as part of the 8:7 filter. Furthermore, samples  $x_i[16 \cdot k + p]$  are 0 for odd values of  $p$  and samples  $x_q[16 \cdot k + p]$  are 0 for even values of  $p$ , and they are not shown in Fig. 5.

The required 8:7 change (from  $P=16$  to  $P=14$ ) in the sample rate follows the steps depicted in Fig. 6. First, the samples are upsampled by a factor of 7, then filtered to avoid aliasing, and finally downsampled by a factor of 8. As shown in Fig. 6, in this process each batch of 16 input samples produces 14 output samples. The upsample, low-pass filter (LPF) and downsample stages can be efficiently implemented by means of a polyphasic structure [17], which profits from the fact that 6 out of 7 samples at the input of the filter are 0, and 7 out of each 8 samples at the output are discarded. In our case, we use a FIR LPF of order 48 that is designed using the Parks-McClellan algorithm [18]. The passband frequency of the filter is set to the maximum frequency of the modulated signal in baseband and the stopband frequency is set to the frequency where the first replica after interpolation by 7 appears. The 49 coefficients are split in 7 subfilters, shown in Fig. 7 as  $\tilde{h}_k[n]$ . The 7 subfilters can provide in parallel 7 consecutive samples of the output  $\hat{y}$ . In Fig. 7 only the output samples that appear in a box have to be computed. This means that when working in parallel, as is our case, for each batch of 16 input samples,  $16 \times 7 = 112$  samples would be created (16 samples per subfilter), but thanks to the downsampling step each subfilter only computes 2

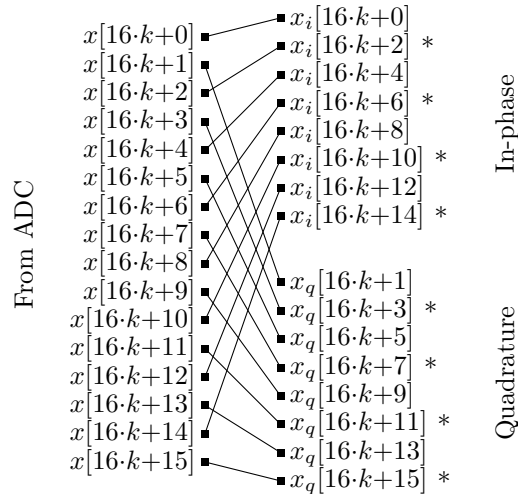


Fig. 5. Simplified  $f_s/4$  mixer.  $k$  denotes the clock cycle number. Samples with the asterisk mark should be negated.

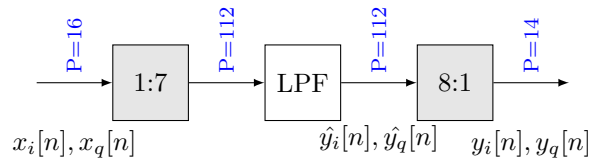


Fig. 6. 8:7 Rational rate change filter.

output samples. It should be noted that  $112/8=14$ , which is the number of parallel samples at the output of the resampler.

As a further simplification, it should be noted that since half the samples at the mixer output are 0, the subfilters of the in-phase branch only employ 4 of 7 coefficients (3 of 7 in the quadrature case): note that Fig. 7 shows the non used in-phase coefficients canceled, and also note that the canceled coefficients are a different set for the quadrature resampler. Moreover, since 1 out of 4 samples coming from the mixer must be negated, instead of changing the sign of the samples, the sign of the coefficients is changed and this operation requires no hardware. Therefore, only 4 and 3 multipliers are employed to compute each one of the 14 parallel samples of the resampler block for the in-phase and quadrature branches, respectively. Fig. 8 shows the implementation of one of the subfilters for the in-phase branch.

### B. Matched filter

The RRC matched filter has a roll-off factor of 0.14 and a span of 20 symbols. The input of the filter has a rate of 2sps and the output has 1sps. Therefore the filter has  $2 \cdot \text{span} + 1 = 41$  coefficients and decimates the data with a 2:1 rate change. Due to the fact that  $P=14$  samples are received in parallel at the filter input, it has to compute  $P=7$  parallel output samples, so 7 parallel filters are required. The hardware implementation schematic of each filter is



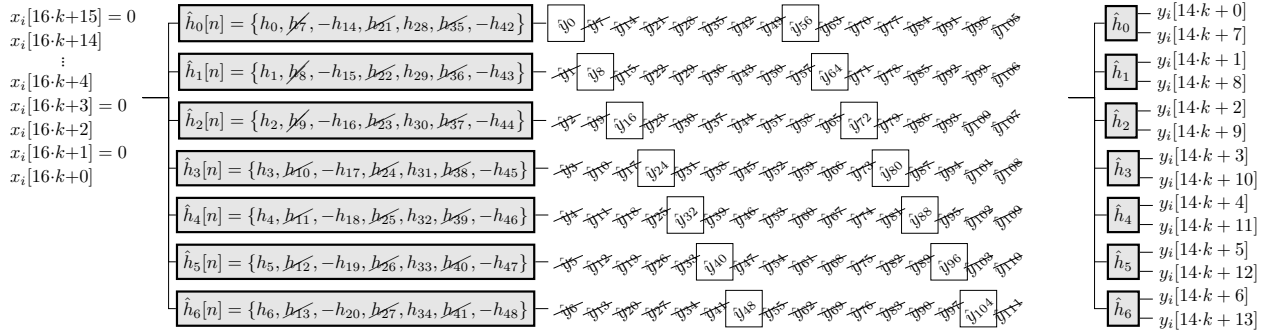


Fig. 7. Polyphase structure for the LPF filter of the resampler block for the in-phase branch.  $k$  denotes the clock cycle number. Note that for the quadrature branch each subfilter only requires 3 coefficients.

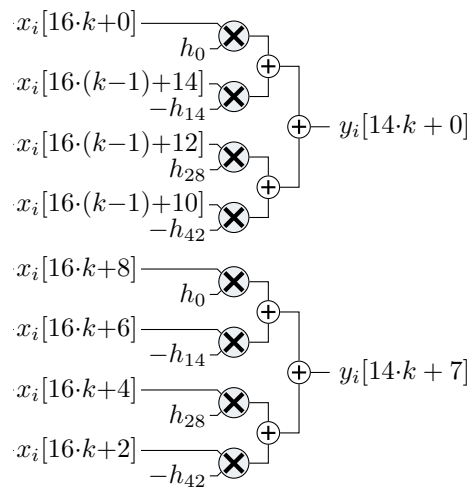


Fig. 8. Subfilter  $\hat{h}_0[n]$  of the in-phase branch.  $k$  denotes the clock cycle number.

shown in Fig. 9a. The 41-tap filter is broken down into 13 3-tap filters and one 2-tap filter (blocks c in Fig. 9) with direct cascade structure followed by a tree adder, as shown in Fig. 9a. Each one of the 7 parallel filters requires 14 cells, therefore, a total of  $14 \times 7$  cells are required.

### C. Frame synchronization

In order to detect the exact instant of the beginning of a frame, we include a sequence in the preamble that is based on Zadoff-Chu sequences [15], which have good autocorrelation properties:

$$a[k] = e^{-\frac{j \cdot \pi R \cdot k \cdot (k+1)}{M}} \quad (k = 0, 1, \dots, M-1), \quad (2)$$

where  $M$  and  $R$  are relative prime, and  $M$  is odd. Since the correlation is a computation that requires as many complex multiplications as the size of the sequence, and as in our case  $P=14$  correlations must be computed in parallel, we simplify the computation by using the next quantized sequence:

$$\hat{a}[k] = \text{sign}(\text{Re}(a[k])) + j \cdot \text{sign}(\text{Im}(a[k])), \quad (3)$$

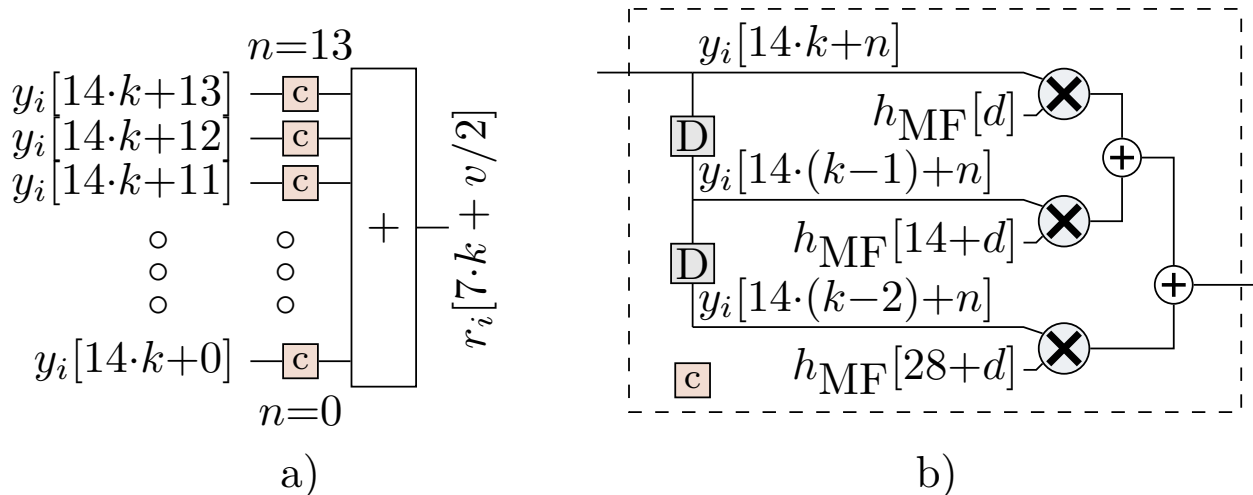


Fig. 9. a) Diagram block of one of the 7 parallel matched filters.  $v$  values for the different parallel filters are  $\{0, 2, 4, 6, 8, 10, 12\}$ . b) Basic cell for the implementation of the matched filters. Offset  $d$  is  $\text{mod}(v-n, 14)$  for the  $n$ -th cell of the filter. Those cells with  $n > v$  have an additional delay at their output.  $h_{\text{MF}}$  are the coefficients of the matched filter.  $k$  denotes the clock cycle number.

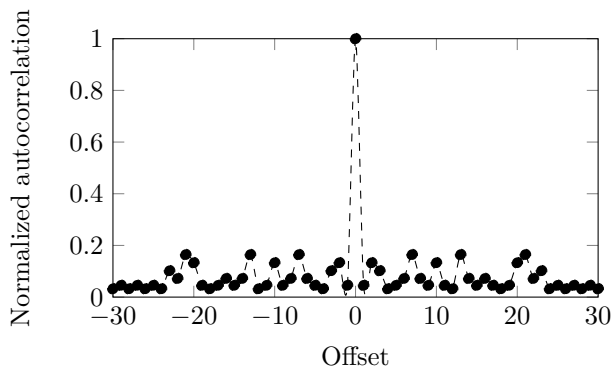


Fig. 10. Autocorrelation of the preamble sequence  $\hat{a}[k]$  ( $M=31$  and  $R=3$ ) selected for frame synchronization.

so the correlation can be computed without multipliers. Under this constraint, we searched a combination of  $R$  and  $M$  that with a small  $M$  (to reduce complexity) gave a good ratio between the maximum and the second maximum of the correlation between received  $\hat{a}[k]$  and  $\hat{a}[k]$ , under noise, phase error and timing error conditions. The sequence included in the preamble of the frame (see field FS in Fig. 4) has  $M=31$  and  $R=3$ . The autocorrelation of the selected  $\hat{a}[k]$  is displayed in Fig. 10.

As shown in Fig. 3, the output of the correlator (named Frame Sync) is used to synchronize the reception of the frame. The inputs to the correlator are the samples of the in-phase and quadrature branches of the receiver, interpreted as complex numbers. Fig. 11 shows the schematic of the  $P=14$  parallel frame synchronization correlator where each one of the blocks in the first stage (labeled as  $\langle \cdot, \cdot \rangle$ ) computes the dot product between 31 complex input samples and the  $\hat{a}[k]$  coefficients. As explained above, since the real and imaginary values of those coefficients

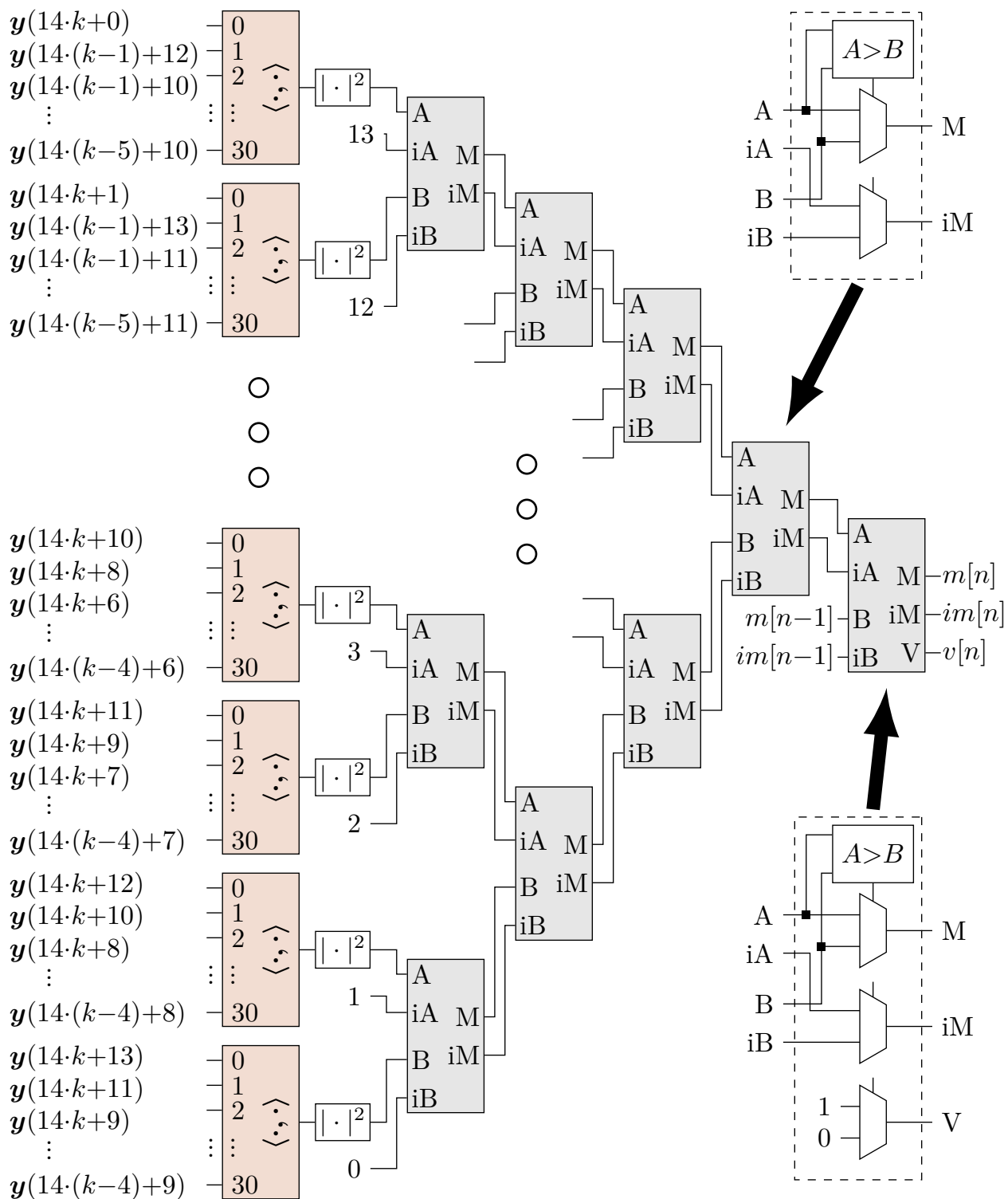


Fig. 11. Implementation schematic of the  $P=14$  parallel frame synchronization correlator.

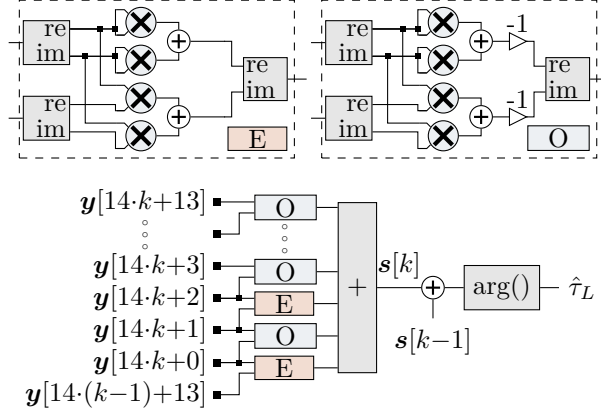


Fig. 12. Timing detector.  $k$  denotes the clock cycle number.

are  $\pm 1$ , no multipliers are required and these blocks are basically two trees of adders (one for the real values and one for the imaginary values). The output of these blocks is a complex value whose absolute square is computed in the next stage. As the desired output of the correlator is the instant when the maximum is detected, a tree of comparators propagates the maximum between couples of absolute squares and its index. The index is the number  $\{0, 1, 2, \dots, 13\}$  of the correlator output in the  $P=14$  batch of computations. Finally, as the search of the maximum must work across several batches of 14 samples, the maximum at the tree output is compared with the previous maximum ( $m[n-1]$ ) to determine the current maximum  $m[n]$  and its index.

#### D. Symbol/timing synchronization

The symbol sync block (see Fig. 3) uses the non-data-aided algorithm for timing estimation proposed by Lee [16], which works at 2 sps:

$$\hat{\tau}_L = \frac{1}{2\pi} \arg \left\{ \sum_{n=1}^{2L} \left[ |y[n]|^2 e^{-jn\pi} + \operatorname{Re} \left[ y[n] y^*[n-1] e^{-j(n-0.5)\pi} \right] \right] \right\} \quad (4)$$

where  $\hat{\tau}_L$  is the normalized timing estimation and  $L$  is the number of symbols involved in the computation. The timing estimation block captures its output only once, at a time determined by the output of the frame synchronization block.

Eq. 4 is implemented as shown in Fig. 12: the inputs to this block are the samples from the 8:7 resamplers of both processing branches, in-phase and quadrature, interpreted as 14 complex numbers that are computed in parallel. In order to simplify the implementation of this block,  $L$  is selected as 14, so the estimation can be computed in 2 clock cycles. It should be noted that the multiplication by the exponential terms in Eq. 4 is just either multiplying by 1 or  $-1$ .  $L$  terms of the sum are computed in one cycle, therefore the complete sum is obtained as the addition of the terms evaluated in 2 clock cycles. Finally, the computation of  $\arg(a+jb)$  is based on a LUT where  $a$  and  $b$

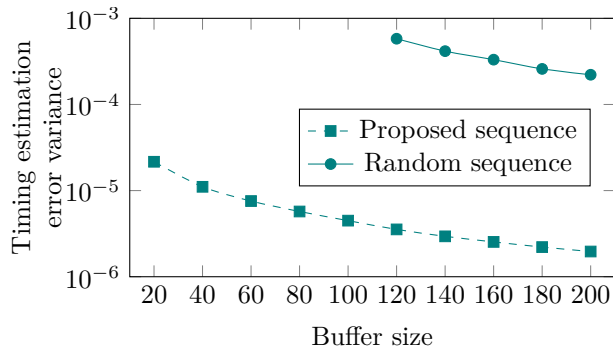


Fig. 13. Variance of the symbol synchronization estimation error for the proposed sequence and for random sequences as a function of the size of the buffer. SNR=20 dB.

are concatenated to form the address input (6+6 bits) of the table. Both  $a$  and  $b$  are pre-scaled by the same power of 2 by means of 2 multipliers in order to use as many significant digits as possible.

Although Eq. 4 is non-data aided and, therefore, can be used with arbitrary input data, we propose to do the estimation using a transmitted sequence that alternates two symbols with  $\pi$  radians phase difference. Assuming a normalized  $[0, 1)$  timing estimation, the proposed sequence reduces the estimation error by two orders of magnitude when compared with random symbol sequences, as shown in Fig. 13 for different buffer sizes ( $2L$ ) and a Signal-to-Noise Ratio (SNR) of 20 dB. This sequence allows the receiver to obtain a valid timing estimation with a reduced buffer and latency. Therefore, in the preamble  $2L = 28$  symbols are included (see Fig. 4) to be used as input for the symbol synchronization block. This length ensures that all the samples in two consecutive clock cycles (*i.e.* 28 samples) belong to this part of the preamble. Since according to Fig. 13, the variance of the estimation error for  $2L = 28$  samples is comparable to rounding to 6 bits at SNR=20 dB, the estimation error is low enough for our design, since we only use the 5 most significant bits of the estimation, as is explained below.

Only the 5 most significant bits out of the detector are employed. The most significant one is used to decide if the received samples have to be delayed by 1 sample (half symbol period) before entering the RRC filter (see Align block in Fig. 3), and the rest are used to select 1 out of 16 different versions of the RRC filter coefficients, which implement a fractional interpolation between two consecutive samples. The necessary coefficients for those 16 filters are stored in a set of 41 ROMs that contain the 16 versions of each coefficient.

### E. Linear equalizer

The last stage of the receiver is a symbol-spaced linear feed-forward equalizer that processes the output samples from the matched filter to reduce the inter-symbol interference caused by both the channel and the filters along the data path. Once equalized, samples can be demapped to obtain the received bits. Following [9], the equalizer order was set to  $L_{Eq} = 40$ .

As shown in Fig. 4, the last part of the preamble contains five copies of a Zadoff-Chu sequence  $\mathbf{c}$  of length  $M=31$  intended to estimate the channel impulse response  $\mathbf{h}$ , with  $R=5$ . The first sequence is sent as a cyclic prefix

TABLE I  
RECEIVERS DOWNCONVERTER COMPUTATIONAL LOAD

SCM system	MF span	CL <sub>MF</sub> (Gmult/s)	M+R order	CL <sub>M+R</sub> (Gmult/s)	Eq order	CL <sub>Eq</sub> (Gmult/s)	Total CL (Gmult/s)
HC [9]	350	1752.5	-	-	200	2000	5505
9/4 [9]	20	135.5	20	33.3	40	364.4	702.2
16/7	20	89.7	48	15.3	40	358.8	568.8

of the other four, so at the receiver we can work with circular convolution. Before channel estimation, the last four received sequences are averaged to reduce the noise, giving sequence  $\mathbf{r}$  of length  $M$ . The received sequence can be expressed as:

$$\mathbf{r} = \mathbf{C}\mathbf{h} + \mathbf{w}, \quad (5)$$

where  $\mathbf{w}$  is a vector containing the channel noise and  $\mathbf{C}$  is a matrix whose columns are formed with delayed versions of  $\mathbf{c}$  to represent the circular convolution with the channel impulse response (of length  $L$ ):

$$\mathbf{C} = \begin{bmatrix} c[0] & c[M-1] & \dots & c[M-L] \\ c[1] & c[0] & \dots & c[M-L+1] \\ \vdots & \vdots & \ddots & \vdots \\ c[M-1] & c[M-2] & \dots & c[M-L-1] \end{bmatrix}_{M \times L} \quad (6)$$

Thanks to the autocorrelation property of the Zadoff-Chu sequences we have:

$$\mathbf{C}^H \mathbf{C} = M \mathbf{I}_{L \times L}, \quad (7)$$

where  $(\cdot)^H$  denotes Hermitian transpose and  $\mathbf{I}$  is the identity matrix. Then, the channel impulse response can be estimated as:

$$\hat{\mathbf{h}} = \frac{1}{M} \mathbf{C}^H \mathbf{r} = \frac{1}{M} \mathbf{C}^H \mathbf{C} \mathbf{h} + \frac{1}{M} \mathbf{C}^H \mathbf{w} \quad (8)$$

giving:

$$\hat{\mathbf{h}} = \mathbf{h} + \frac{1}{M} \mathbf{C}^H \mathbf{w} \quad (9)$$

Once the channel impulse response has been estimated, the coefficients of the equalizer can be obtained [19] and applied.

#### IV. EVALUATION OF COMPUTATIONAL LOAD

In this section we report the computational load of the proposed receiver, defined as the number of multiplications per second (mult/s). First, for the sake of comparison with reference [9], we report the values for the down-converter blocks. It should be noted that all the blocks of the receiver work at a frequency of  $f_{clk} = 312.5$  MHz.

The matched filter computes  $P=7$  samples in parallel. Since it has  $2\cdot\text{span}+1$  coefficients, and two filters are required (for the in-phase and quadrature branches), the total number of multipliers is  $2\cdot P\cdot(2\cdot\text{span}+1)=574$ . The computational load of each filter is:

$$CL_{MF} = P\cdot(2\cdot\text{span}+1)\cdot f_{clk} = 7\cdot 41\cdot f_{clk} = 89.7 \text{ Gmult/s.}$$

As was explained in Section III-A, the mixer plus resampler block computes  $P=14$  samples in parallel. Each one of the  $P$  computations requires  $(N_{MR}+1)/(7\cdot 2)=3.5$  multiplications, where  $N_{MR}$  is the order of the filter. In this case, the two filters for the in-phase and quadrature branches require a total of  $2\cdot P\cdot(N_{MR}+1)/(7\cdot 2)=98$  multipliers. The computational load of each branch is:

$$CL_{M+R} = P\cdot\left(\frac{N_{MR}+1}{7\cdot 2}\right)\cdot f_{clk}=14\cdot 3.5\cdot f_{clk}=15.3 \text{ Gmult/s.}$$

The computational load of the equalizer is:

$$CL_{Eq} = P\cdot(L_{Eq} + 1)\cdot 4\cdot f_{clk} = 7\cdot 41\cdot 4\cdot f_{clk} = 358.8 \text{ Gmult/s,}$$

where factor 4 accounts for the multiplications being complex.

In Table I, we compare the computational load of the down-converter stage of the proposed receiver with the systems analyzed in [9]. As can be seen in the results, the simplifications we propose reduce the computational a 19% when compared with the 9/4-SCM, a reduction that comes mostly from a matched filter that works with 2 sps instead of 3 sps. When compared with the HC-SCM, the computational load is reduced by 90%.

The symbol synchronization block requires  $N_{TS}=4$  multiplications for the computation of one term of the sum in Eq. 4, and 2 additional multipliers are used in the pre-scale of the inputs of the  $\arg()$  function. Therefore, the total amount of multipliers required for  $P=14$  parallel computations is  $P\cdot N_{TS}+2=58$ . Its computational load is:

$$CL_{TS} = (P\cdot N_{TS} + 2)\cdot f_{clk} = (14\cdot 4 + 2)\cdot f_{clk} = 18.1 \text{ Gmult/s.}$$

The correlator requires  $N_{CO}=2$  multipliers to compute the squared module of each one of the  $P=14$  outputs. Therefore, a total of 28 multipliers are required for this computation. The computational load for the correlator is:

$$CL_{CO} = P\cdot N_{CO}\cdot f_{clk} = 14\cdot 2\cdot f_{clk} = 8.8 \text{ Gmult/s.}$$

It should be noted that our correlator is a simplified version that requires no multipliers, as explained in Section III-C. If the original Zadoff-Chu coefficients (see Eq. 2) were used instead of the proposed simplified version (see Eq. 3), the computational load of this block would be increased by:

$$\Delta CL_{CO} = P\cdot M\cdot \Delta N_{CO}\cdot f_{clk} = 14\cdot 31\cdot 4\cdot f_{clk} = 542.5 \text{ Gmult/s,}$$

where  $M$  is the length of the Zadoff-Chu sequence and  $\Delta N_{CO}$  is the number of multipliers required to compute a complex multiplication. The total amount of required multipliers would be increased by  $P\cdot M\cdot \Delta N_{CO} = 1736$ .

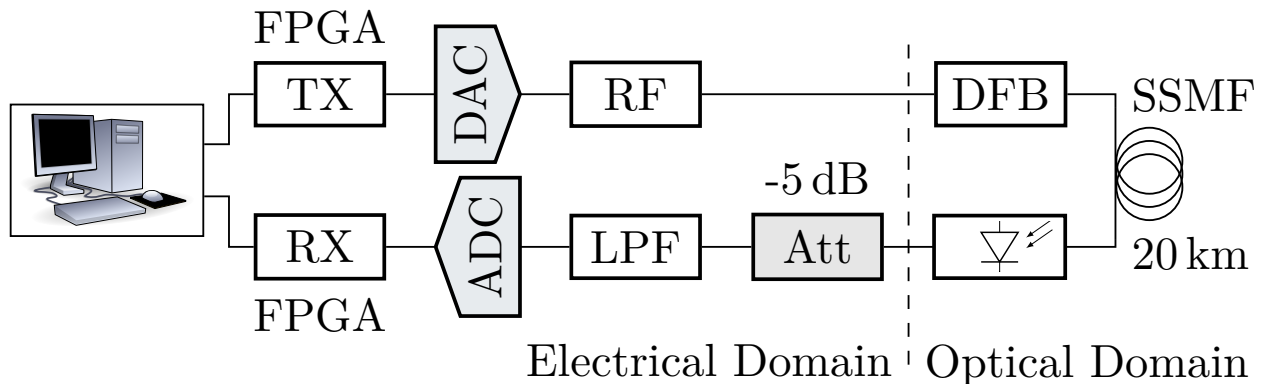


Fig. 14. Experimental setup.

## V. EXPERIMENTAL RESULTS

### A. Experimental setup

The experimental setup for the proposed IM/DD optical communication link is shown in Fig. 14. The OVS-HCM signal was generated using Matlab with 16/7 sps and a roll-off factor of 0.14. Moreover, a pre-emphasis was added to compensate for the digital-to-analog (DAC) frequency response, with a maximum amplification of 6 dB (at 2.5 GHz). The signal was then sent to the VC707 Xilinx FPGA evaluation board, which includes the Euvis DAC MD657B (12 bit) with a sampling rate of 5 GSa/s and a 2 GHz 3 dB bandwidth. After the DAC converter, the analog signal was amplified and converted to the optical domain with a low cost directly modulated laser (DML) whose wavelength and optical power are 1550 nm and 4.3 dBm, respectively. Since this signal was transmitted through 20 km of SSMF, with an attenuation of 0.19 dB/km, at the fiber output the signal had 0.2 dBm optical power. In the receiver part of the communication link, the first step was a InGaAs PIN photodiode (with a responsivity of 0.94 A/W at 1550 nm) that converted the received signal into the electrical domain. The whole electrical to optic and optic to electrical system had an RF bandwidth of 3 GHz. The amplitude of the photodetected signal was then adjusted with an attenuator to avoid saturation at the input of the ADC. The signal was filtered with a low-pass (LPF) anti-aliasing filter with a 3 dB bandwidth of 2.35 GHz, a maximum insertion loss of 1.8 dB and a minimum rejection of 40 dB for frequencies above 2.65 GHz. A 10-bit ADC, model EV10AQ190 from E2V, was then used to sample the signal at 5 GSa/s (with a 3.2 GHz 3 dB bandwidth) and the data was delivered to a second VC707 Xilinx FPGA evaluation board. Finally, the captured signal was sent to a PC, where the received signal was processed offline in Matlab.

### B. Measurement results

Table II presents the measurements of BER (bit error rate) and EVM (Error Vector Magnitude) for the 2.25 sps OVS-SCM system from [9] and the proposed 2.28 sps OVS-SCM system. 2,265,000 symbols were used to measure the BER, which were generated using the Mersenne Twister method with the Matlab function “rng”. It can be seen that the proposed system has as good performance as the one in [9]: for several QAM modulation orders both



solutions obtain similar EVM values, and BER values are lower than the hard decision forward error correction (HD-FEC) threshold of  $3.8 \cdot 10^{-3}$  [20]. In this table, it is also shown the achieved data rate: as expected the 2.25 sps solution gives a bit higher throughput.

Fig. 15 shows the measurements of BER vs. the received optical power (ROP) for QAM signals. Values for 0 dBm are next to those shown in Table II (measured at 0.2 dBm). The HD-FEC threshold are at  $-1$ ,  $-5.8$  and  $-8$  dBm for 64/128/256-QAM, respectively. It should be noted that if the main distortion in the system were the receiver noise, the three curves would exhibit the same slope. But, as the transmission channel distorts the signal and an equalizer is required, for lower modulation orders the performance is better.

The transmitted and received 64-QAM signal spectra measured with a spectrum analyzer after the DAC and before the ADC, respectively, are shown in Fig. 16. It can be seen how the transmitted signal with a roll-off  $\beta=0.14$  is centered and confined in the 2.5 GHz bandwidth. At the receiver the antialiasing filter cancels out the out-of-band signal to avoid interference. Once demodulated by the proposed receiver, the obtained constellation diagram is represented in Fig. 17(a), where the clear constellation points are related with the low measured EVM.

As commented previously, this proposal not only reduces the computational load of the previous one in [9], but also presents an implementable architecture of the whole receiver. Related with a real implementation, next subsection discusses, by means of measurements, the problem of clock misalignment between transmitter and receiver.

TABLE II  
THROUGHPUT, BER AND EVM RESULTS

RX	Mod. M-QAM	Eq.	span	Thr. (Gb/s)	BER	EVM %
	64			13.3	$3 \cdot 10^{-6}$	4.6
9/4 SCM	128	40	20	15.6	$6.5 \cdot 10^{-5}$	4.5
[9]	256			17.8	$2.3 \cdot 10^{-3}$	4.5
	64			13.1	$1 \cdot 10^{-6}$	4.1
16/7 SCM	128	40	20	15.3	$7.8 \cdot 10^{-5}$	4.1
	256			17.5	$2.6 \cdot 10^{-3}$	4.1

### C. Performance under sampling clock frequency offset

All the results presented in the sections above were obtained with a receiver working with a main clock that matches the one used by the transmitter. Nevertheless, in a real case these two clocks will be different, with a difference that is usually expressed as parts per million (ppm). When this sampling clock frequency offset (SCFO) exists, two problems in the receiver arise: there will be a progressive rotation of the received constellation and the samples at the output of the matched filter will progressively shift in time relative to the symbol time that is optimum for the decision. It should be noted that since the roll-off factor of the modulation scheme is close to zero, timing errors have a dramatic effect in the EVM. In a non-parallel receiver, the drift in the timing of the samples can be counteracted with a dynamic resampling block based on a fractional interpolator, but in a parallel receiver, as is

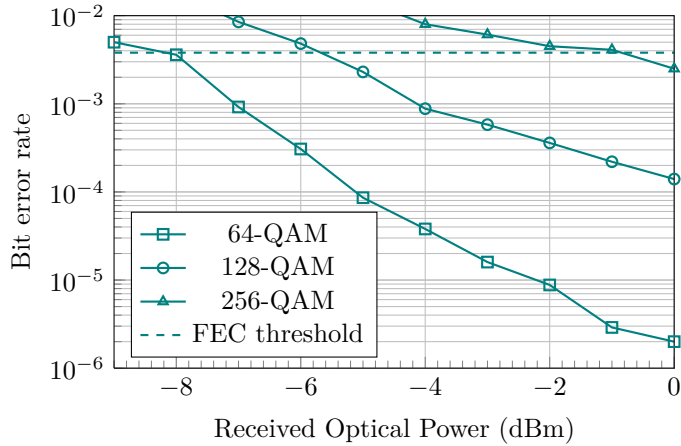


Fig. 15. Bit error rate vs. received optical power.

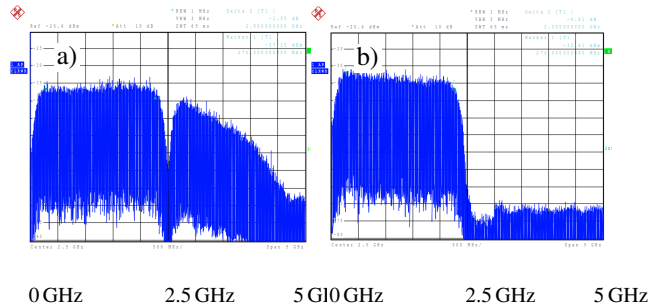


Fig. 16. 64-QAM 16/7 OVS-SCM Signal Spectrum after DAC (a) and after the anti-aliasing filter (b).

our case, the overflows and underflows of the fractional interval ( $\mu$ ) are almost impossible to handle. Other authors (see [21]) have proposed systems where the SCFO problem is removed by adjusting the ADC clock frequency to the frequency of the received signal. Another possibility (see [22], [23]) is to transmit data with a nominal baud rate slightly below the one of the receiver, so that only underflows in the  $\mu$  can happen and a reception buffer can discard samples when the underflow occurs. The design of the buffer in this approach sets a limit to the number of underflows that can be managed. This specification limits the size of the payload that can be transmitted. With this solution, the mismatch between the transmitter and receiver clocks still causes a progressive rotation of the constellation that must be addressed. For example, in [22], [23] this frequency and phase offset is compensated by means of a Viterbi & Viterbi algorithm.

Fig. 17 a) shows the measured scatter plot of the received symbols without SCFO. For a payload of 5000 symbols with a SCFO of 10 ppm, the scatter plot is the one shown in Fig. 17 b), where the symbols are progressively both rotated and more scattered. As shown in Fig. 18, for 0 ppm the EVM is close to 4% but for 10 ppm it increases progressively from 4% to 19% at the end of the 5000-symbol payload. As can be seen, for 20 and 40 ppm the measured final EVM is close to 38% and 75%, respectively. In Fig. 17 c), if the rotation in the constellation is correctly estimated, it can be seen how it can be counteracted just by multiplying the symbols by the appropriate

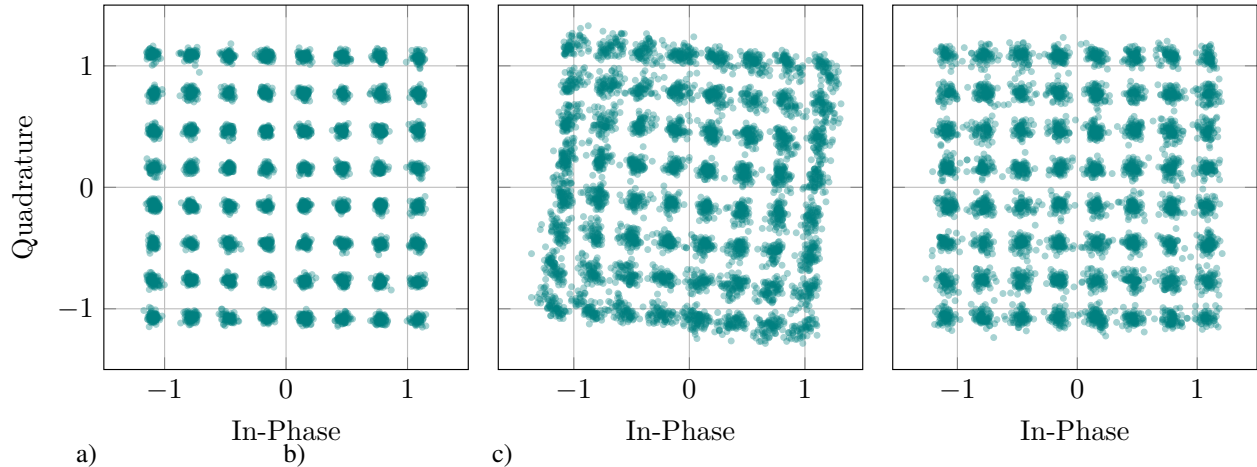


Fig. 17. Measured scattering diagram of 64-QAM signal with 5000 payload symbols and receiver with SCFO of (a) 0 ppm, (b) 10ppm and (c) 10 ppm with rotation compensation.

unitary complex numbers. As shown in Fig. 18, when this solution is applied, the EVM still deteriorates due to the drift in the sample times, but an acceptable EVM can be obtained with longer payloads. For example, for 10 ppm, a maximum EVM of 8.5% is obtained with a payload of 5000 symbols. This result would satisfy the HD-FEC threshold for 64-QAM.

One solution we suggest to handle the SCFO problem is to limit the size of the frame to around 4100 symbols when there is a clock difference of 10 ppm. To estimate the rotation at the receiver, the transmitter would send 2 modified Zadoff-Chu sequences (see Eq. 3 with  $M=31$  and  $R=3$ ), one before and one after the payload. By using the angle of the correlation with both sequences, the rotation of the constellation can be estimated at the receiver. After that, this would be used to rotate the received symbols and so to counteract the effect of the SCFO in the input samples, which should have been stored in a FIFO memory. Our experiments show that by using this scheme, for a payload of 4065 symbols and a distance between the modified Zadoff-Chu sequences of 4096 symbols, the EVM at the end of the payload is significantly below 8.5% (for a 95% confidence interval). As explained above, this outcome would satisfy the HD-FEC threshold for 64-QAM.

## VI. FPGA IMPLEMENTATION RESULTS

The proposed receiver architecture is designed to process the data captured by the 10-bit EV10AQ190 ADC. It samples at 5 GSa/s the received data and delivers 4 samples in parallel to the FPGA, using 40 LVDS pairs, at a rate of 1.25 Gbps. Each one of the sample flows is further converted from serial to parallel, providing 4 samples in parallel with a sample rate of 312.5 MHz, 4 times lower. Therefore, the target clock frequency for the FPGA implementation is 312.5 MHz, with 16 parallel channels to be processed.

The proposed receiver was modelled using System Verilog language and verified using a Matlab finite precision model. The widths of the datapath were chosen to minimize the use of multipliers while taking full advantage of

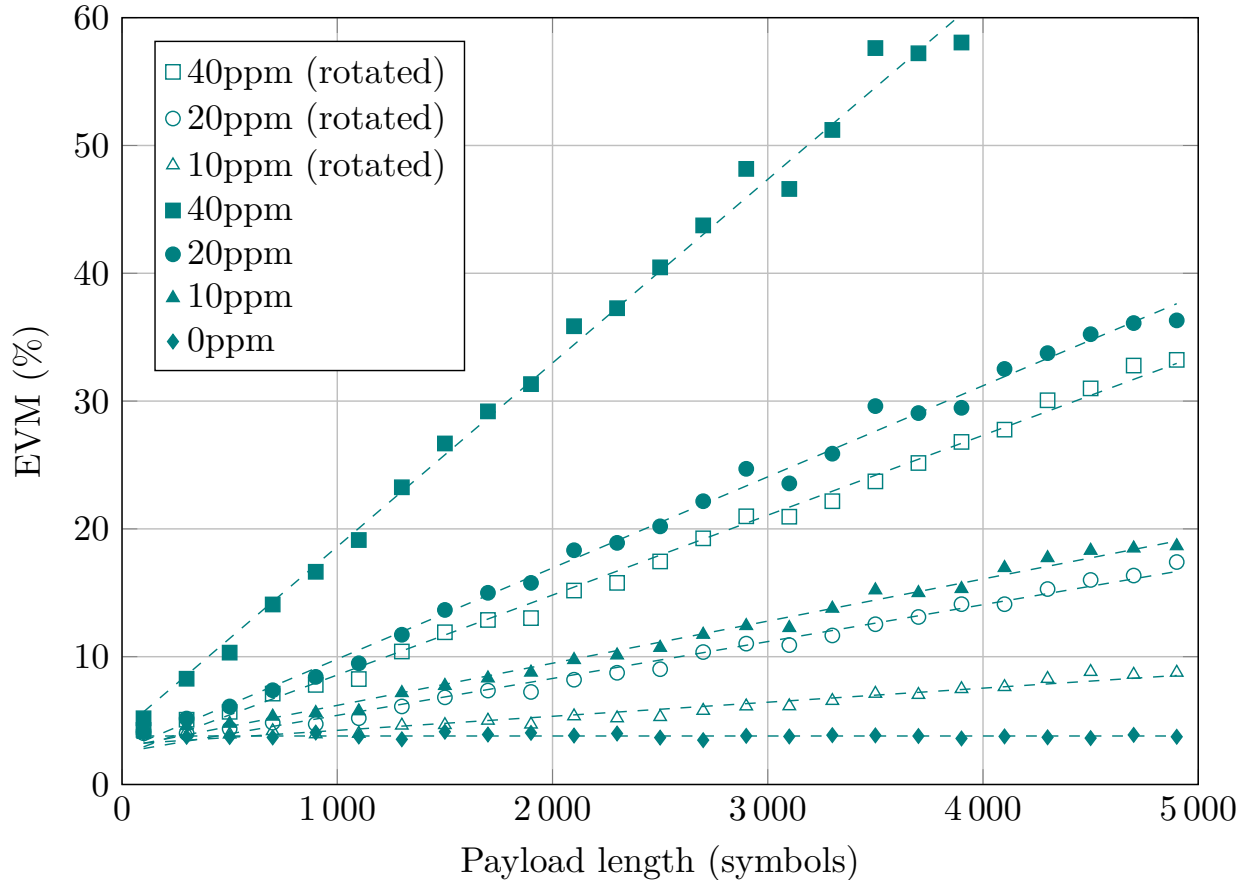


Fig. 18. Measured EVM deterioration due to sampling clock differences between transmitter and receiver. The EVM is measured for each 200 consecutive symbols.

their size ( $25 \times 18$  bits in DSP48 blocks). For example in the filters, the data make use of the 25-bit inputs and the coefficients have a width of 18 bits. The different blocks of the receiver have been implemented following the schematic diagrams showed previously, but with full pipeline, so the target clock frequency could be reached. Extra registers have been added in the design, as required, in order to reduce propagation delays between the operators implemented in the FPGA slices and the DSP48 blocks. The receiver architecture was implemented in a Xilinx Virtex-7 XC7VX485T-2 FPGA using the Xilinx Vivado 2016.3 software tool. The results of the FPGA receiver implementation are shown in Table III, where the use of the chip resources by blocks is detailed. It should be noted that the number of DSP48 blocks obtained matches the predicted (see Section IV) multipliers. The BRAM is required for the implementation of the  $\arg()$  function.

## VII. DISCUSSION

In this section, we discuss how the results from the present work can be extended to achieve higher bit rates, by taking advantage of the improvement in CMOS technology. First of all, let's remember that the design we propose works with a clock frequency of 312.5 MHz, which is the result of using a  $P=16$  parallelization and an ADC that

TABLE III  
FPGA IMPLEMENTATION RESULTS

Block	LUTs	FFs	Slices	BRAMs	DSP48
Mixer/Resampler 8:7	3008	3099	1123	0	98
RRC	44578	23133	14062	0	574
Frame sync.	16478	19523	6114	0	28
Symbol sync.	877	1314	490	1	58
Buffer+align	4640	2808	2460	0	0
Total receiver	69581	49877	24249	1	758

provides 5 GSa/s. If higher bit rates are desired, a faster ADC must be used, and the FPGA device must be able to increase its clock frequency and/or must have enough resources to support a higher parallelization degree.

Table IV shows the results of implementing the receiver architecture (with  $P=16$ ) on the target device (a Virtex 7 device) and on two newer devices (a 16 nm Virtex-7 UltraScale+, xcvu9p-fsgd2104, and a 7 nm Versa Adaptive Compute Acceleration Platform, xcvc1902-vsua2197-3HP). As can be seen, the LUT and flip-flop count for the implementation is similar for all 3 devices. In all cases, 1 BRAM and 758 DSP48 blocks are used. In addition to the number of resources, the table shows the maximum clock frequency ( $f_{clk\_max}$ ) achieved by the proposed architecture on each device, along with the maximum sample rate ( $f_{s\_max}$ ) and throughput (Thr) achievable assuming 128-QAM is employed. These results have been obtained using the Xilinx Vivado 2021 software tool. The main difference is a near 50% increase in the maximum clock frequency, which would allow for an increase from 5 GSa/s up to 7.6 GSa/s (for the Versal device). Logically, if a much higher rate is desired, a higher degree of parallelization must be used, which means that more resources are required.

TABLE IV  
FPGA IMPLEMENTATION RESULTS FOR DIFFERENT DEVICES

Device	Process (nm)	LUTs	FFs	$f_{clk\_max}$ (MHz)	$f_{s\_max}$ (GHz)	Thr (Gbps)
Virtex7	28	69581	49877	312.5	5.0	15.3
Virtex7US+	16	71888	49107	454.5	7.3	22.3
Versal	7	81878	49746	476.2	7.6	23.3

For example, a receiver with parallelization  $P=64$  and a clock frequency of  $f_{clk}=312.5$  MHz can be implemented in a Virtex UltraScale+, receiving the data from a 20 GHz ADC (10 GHz bandwidth). In this case, a throughput of 52.5 Gbps could be obtained assuming transmission of 64-QAM signals.

Although FPGA devices are constantly increasing the number of available resources thanks to the reduction in the CMOS process, the fact that higher degrees of parallelization are presumably required highlights the importance of using designs optimized for parallelization. For example, inherent recursive algorithms, like the ones used for adaptive equalizers or synchronization stages that rely on feedback loops, are quite problematic because, not only

the parallelization increases the hardware resources, but also the critical path increases with the parallelization. Therefore, a good selection of architecture and algorithms can help to achieve a considerable reduction in the amount of required resources, which logically allows for an even greater degree of parallelization and, as a consequence, a higher throughput.

### VIII. CONCLUSION

In this paper, we propose the design and hardware architecture of an OVS-SCM receiver, which takes 16 samples in parallel from a 5 GHz analog-to-digital converter. Specifically, we present an efficient solution for the frame detection block, based on a modified Zadoff-Chu sequence, that requires no multipliers; a joint architecture of the  $f_s/4$  mixer and resampler blocks that greatly reduces their complexity; a matched filter, working at 2 samples per symbol, that also performs the fractional interpolation of the timing synchronization block; and a feed-forward timing estimator block, based on a non-data-aided algorithm in the receiver and the inclusion of a specific set of 28 symbols in the preamble of the frame. This proposal, when compared to the HC-SCM receiver, reduces the computational load of the downconverter stages by 90%. Additionally, a solution is proposed to overcome the offset of the sampling clock frequency. We show that 64-QAM can be transmitted accomplishing the HD-FEC threshold with a clock difference of 10 ppm. Finally, results of the FPGA implementation are given to demonstrate the suitability and feasibility of our proposal.

### ACKNOWLEDGMENT

We thank Andrés Suárez-González for his help with the statistical interpretation of the data.

### REFERENCES

- [1] H. Sun et al., "800G DSP ASIC Design Using Probabilistic Shaping and Digital Sub-Carrier Multiplexing," *Journal of Lightwave Technology*, vol. 38, no. 17, pp. 4744-4756, 1 Sept.1, 2020.
- [2] M. Yin, W. Wang, D. Zou, Z. Luo, Q. Sui, X. Yi, F. Li, and Z. Li, "DMulti-band DFT-S 100 Gb/s 32 QAM-DMT transmission in intra-DCI using 10 G-class EML and low-resolution DAC," *Opt. Express*, vol. 30, no. 18, pp. 32742-32751, 2022.
- [3] K. Zhong, X. Zhou, J. Huo, C. Yu, C. Lu, and A. P. T. Lau, "Digital Signal Processing for Short-Reach Optical Communications: A Review of Current Technologies and Future Trends," *Journal of Lightwave Technology*, vol. 36, no. 2, pp. 377-400, 2018.
- [4] J. S. Bruno, V. Almenar, J. Valls, and J. L. Corral, "Real-time 20.37 Gb/s optical OFDM receiver for PON IM/DD systems," *Optics Express*, vol. 26, no. 15, pp. 18 817-18 831, 2018.
- [5] M. Chen, X. Xiao, Z. R. Huang, J. Yu, F. Li, Q. Chen, and L. Chen, "Experimental Demonstration of an IFFT/FFT Size Efficient DFT-Spread OFDM for Short Reach Optical Transmission Systems," *Journal of Lightwave Technology*, vol. 34, no. 9, pp. 2100-2105, 2016.
- [6] S. Amiralizadeh, A. Yekani, and L. A. Rusch, "Discrete Multi-Tone Transmission with Optimized QAM Constellations for Short-Reach Optical Communications," *Journal of Lightwave Technology*, vol. 34, no. 15, pp. 3515-3522, 2016.
- [7] J. Tang, J. He, D. Li, M. Chen, and L. Chen, "64/128-QAM half-cycle subcarrier modulation for short-reach optical communications," *IEEE Photonics Technology Letters*, vol. 27, no. 3, pp. 284-287, 2015.
- [8] J. C. Cartledge and A. S. Karar, "100 Gb/s intensity modulation and direct detection," *Journal of Lightwave Technology*, vol. 32, no. 16, pp. 2809-2814, 2014.
- [9] A. Pérez-Pascual, J. S. Bruno, V. Almenar, and J. Valls, "A computational efficient Nyquist shaping approach for short-reach optical communications," *Journal of Lightwave Technology*, vol. 38, no. 7, pp. 1651-1658, 2020.

- [10] H. Wang et al., "Multi-Rate Nyquist-SCM for C-Band 100 Gbit/s Signal Over 50 km Dispersion-Uncompensated Link," *Journal of Lightwave Technology*, vol. 40, no. 7, pp. 1930-1936, 1 April, 2022.
- [11] A. S. Karar and J. C. Cartledge, "Generation and detection of a 56 Gb/s signal using a DML and Half-Cycle 16-QAM Nyquist-SCM," *IEEE Photonics Technology Letters*, vol. 25, no. 8, pp. 757-760, 2013.
- [12] K. Zhong, X. Zhou, Y. Gao, Y. Yang, W. Chen, J. Man, L. Zeng, A. P. T. Lau, and C. Lu, "Transmission of 112 Gbit/s single polarization half-cycle 16-QAM Nyquist-SCM with 25 Gbps EML and direct detection," *European Conference on Optical Communication, ECOC*, vol. 2015-November, pp. 1-3, 2015.
- [13] R. Schmogrow, M. Winter, M. Meyer, D. Hillerkuss, S. Wolf, B. Baeuerle, A. Ludwig, B. Nebendahl, S. Ben-Ezra, J. Meyer, M. Dreschmann, M. Huebner, J. Becker, C. Koos, W. Freude, and J. Leuthold, "Real-time Nyquist pulse generation beyond 100 Gbit/s and its relation to OFDM," *Opt. Express*, vol. 20, no. 1, pp. 317-337, Jan 2012.
- [14] R. Schmogrow, M. Meyer, P. Schindler, B. Nebendahl, M. Dreschmann, J. Meyer, A. Josten, D. Hillerkuss, S. Ben-Ezra, J. Becker, C. Koos, W. Freude, and J. Leuthold, "Real-time nyquist signaling with dynamic precision and flexible non-integer oversampling," *Opt. Express*, vol. 22, no. 1, pp. 193-209, Jan 2014.
- [15] D. Chu, "Polyphase codes with good periodic correlation properties (corresp.)," *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 531-532, 1972.
- [16] S. J. Lee, "A new non-data-aided feedforward symbol timing estimator using two samples per symbol," *IEEE Communications Letters*, vol. 6, no. 5, pp. 205-207, May 2002.
- [17] F. Harris, *Multirate Signal Processing for Communication Systems*. Prentice Hall, 2004.
- [18] J.H. McClellan and T.W. Parks, "A personal history of the Parks-McClellan algorithm," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 82-86, 2005.
- [19] J. Kurzweil, *An Introduction to Digital Communications*. Wiley, 2000.
- [20] I.-T. S. Group, "ITU-T Rec. G.975.1 (02/2004) Forward error correction for high bit-rate DWDM submarine systems," *Standard*, pp. 1-58, 2005.
- [21] N. Kikuchi, T. Yano, and R. Hirai, "FPGA prototyping of single-polarization 112-Gb/s transceiver for optical multilevel signaling with intensity and delay detection," *Journal of Lightwave Technology*, vol. 34, no. 8, pp. 1762-1769, 2016.
- [22] R. M. Ferreira, J. D. Reis, S. B. Amado, A. Shahpari, F. P. Guiomar, J. R. F. Oliveira, A. N. Pinto, and A. L. Teixeira, "Performance and complexity of digital clock recovery for Nyquist UDWDM-PON in real time," *IEEE Photonics Technology Letters*, vol. 27, no. 21, pp. 2230-2233, 2015.
- [23] R. M. Ferreira, J. D. Reis, S. M. Rossi, S. B. Amado, F. P. Guiomar, A. Shahpari, J. R. F. Oliveira, A. N. Pinto, and A. L. Teixeira, "Coherent Nyquist UDWDM-PON with digital signal processing in real time," *Journal of Lightwave Technology*, vol. 34, no. 2, pp. 826-833, 2016.