Propagation of Optical Orthogonal Frequency Division Multiplexed Signals through Real-Time Digital Signal Processing

Shijan Handa, Vaibhav Kharbanda, Rahul Phogat
Electronics & Communication Engineering
Dronacharya college of Engineering, Gurgaon

Abstract- We check into thoroughly the design of a field programmable gate array (FPGA) based optical orthogonal frequency division multiplexing (OFDM) transmitter implementing real time digital signal processing at 21.4 GS/s. The transmitter was utilized to generate 8.34 Gbit/s QPSK-OFDM signals for direct detection. We study the impact of the finite resolutions of the inverse fast Fourier transform (IFFT) cores and the digital-to-analog converters (DACs) on the system performance.

Index Terms- Optical orthogonal frequency division multiplexing, OFDM, FPGA, DSP, IFFT, digital-to-analog converter (DAC), optical transmission.

I. INTRODUCTION

COMMUNICATION networks form the backbone of the 21st century economy and modern society providing rapid access to and exchange of information around the globe. In the last few years, there has been a considerable increase in demand for higher bandwidth and more reliable data and voice services mainly driven by bandwidth-hungry applications such as multimedia and online social networks. This development has put serious strains on communication networks and exposed the limitations of current network architectures Higher capacity next-generation networks, that are low cost and reliable, are needed to satisfy future bandwidth and flexibility requirements. However, a major obstacle to implementing higher bandwidth architectures is the management of the optical signal degradation and therefore it has become necessary to develop novel techniques and modulation formats that are both spectrally efficient and resilient to fiber impairments. One such promising transmission technique is optical Orthogonal Frequency Division Multiplexing (OFDM). This modulation format is widely used in wireless communications such as LTE, Wi-Fi and WiMax and has recently gained a great deal of interest in optical communications from both the industrial and academic communities. OFDM consists of dividing the bandwidth of the channel into many non-interfering (orthogonal) sub-channels which operate at a fraction of the aggregate speed of the main channel and therefore are less prone to fiber distortion. In addition to resilience, OFDM can achieve high spectral efficiency as it allows for the utilization of advanced coding formats such as quadrature amplitude modulation (QAM). Furthermore, the multiband technique in which many OFDM bands can be aggregated within a single optical channel, can relax the speed and bandwidth requirements of the signal converters while providing finer switching granularity and better network service flexibility.

There are two different methods of implementing optical OFDM systems: direct-detection and coherent detection. The transmitter architecture of the two schemes is essentially the same and they only differ in the receiver design. Direct-detection OFDM (DD-OFDM) has the advantage of a simpler and cheaper design as it only requires a single photodiode and digital signal processing (DSP) at the receiver albeit with reduced sensitivity and spectral efficiency. Coherent detection OFDM (CO-OFDM) on the other hand achieves better sensitivity and spectral efficiency but needs a complex and more expensive receiver due to the requirement for phase and polarization tracking. Major advances in term of spectral efficiency, range and resilience have been achieved recently for both schemes. For instance, data rates between 10 and 100 Gbit/s have been experimentally demonstrated using direct-detection OFDM in short, medium and long-haul transmissions. Similarly, 25-120 Gbit/s coherent

OFDM has been implemented in long and ultra long-haul transmissions. WDM and multi-band OFDM experiments with data rates up to 1 Tbit/s and spectral efficiency of 7 bit/s/Hz have been shown. Furthermore, OFDM technology is ideally suited to provide the next generation of 100G Ethernet services in data centers and local area networks and may be scaled to allow for future Ethernet standards such as 400GbE and 1TbE using the multiband multiplexing technique.

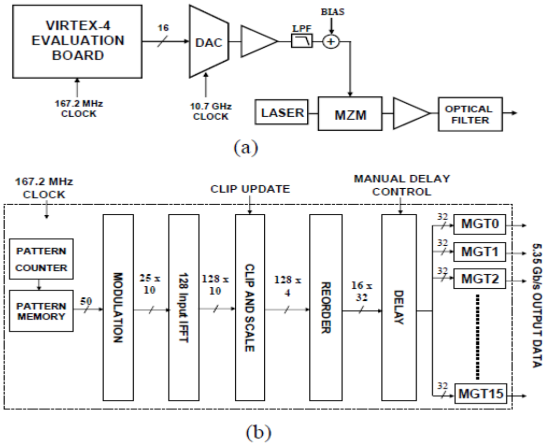
However all this progress has been achieved by using arbitrary waveform generators (AWG) with offline processing at the transmitter and fast sampling oscilloscopes coupled with offline processing at the receiver. This method does not consider hardware limitations such as limited arithmetic resolution, latency and logic requirements. Since OFDM is a primarily digital technique, its adoption at optical communications line-rates has been dictated by the speed of DSP and signal converters (digital-to-analog (DAC) and analog-to-digital (ADC) converters). Recently, multi-gigabit per second DACs and ADCs have become commercially available making the development and implementation of optical OFDM possible. The high-speed DSP implementation can be achieved cost-effectively using field programmable gate arrays (FPGA). The availability of high-speed signal converters and FPGAs has led to the development of the first real-time optical OFDM systems.

We have recently published work on a 21.4 GS/s real-time FPGA-based optical OFDM transmitter. This was used to generate and transmit 8.36 Gbit/s digitally upconverted single sideband directly detected OFDM signals. In this paper we theoretically and experimentally investigate the impact of IFFT and DAC resolutions on the performance of such a system and propose new implementation and design guides. Additionally, we present a transmission experiment over 800 and 1600 km of uncompensated standard fiber using a 10-bit IFFT core and a 4-bit DAC. Section II describes the transmitter hardware design including the DSP part and the analog and optical frontends. Section III explains the IFFT design and algorithms. Section examines the theoretical impact of IFFT and DAC resolutions on the performance of the system using simulation. Section V presents an experimental investigation of the developed theory and also reports a transmission experiment of an 8.36 Gbit/s directly detected OFDM signal over 800 and 1600 km of uncompensated fiber.

II. TRANSMITTER HARDWARE DESIGN

The developed transmitter can be used with both coherent and direct detection receivers and, in this work, it has been adapted to the latter scheme. A guard band is needed in such schemes to ensure that no second order intermodulation products, from the nonlinear mixing of pairs of OFDM sub-channels due to the square-law detector, fall on the used subcarrier frequencies. This guard band can be inserted using analog or digital techniques. The former requires the use of analog mixers to up-convert the OFDM data to a desired frequency band and has the advantage of exploiting the full bandwidth of the DAC. The digital technique is much simpler and consists of inserting virtual carriers to create the guard band albeit with a lower spectral efficiency as only a quarter of the DAC‟s bandwidth can be utilized. The top level design of the transmitter is shown in Fig. The transmitter consists of a DSP block, an analog part and an optical frontend. The hardware and FPGA design used in this work for generating 21.4 GSample/s signals has been previously reported in. Here we describe aspects of the design specific to OFDM generation. The DSP was performed on a Xilinx Virtex-4 (4VFX100) FPGA which was interfaced to a 21.4 GS/s, 4-bit resolution DAC constructed from discrete components (4:1 time division multiplexers, attenuators and a combiner) as described in. The interface between the FPGA and DAC consisted of sixteen serial lines operating at 5.35 Gbit/s using the multi-gigabit transceivers (MGT) on the FPGA. The analog front-end consisted of an amplifier, a Bessel low pass filter with a cut off frequency of 7.5 GHz (a synchronous digital hierarchy (SDH) filter) and a bias. Electrical I/Q mixing was not considered in this work. The amplifier output bias and gain were adjusted to optimize transmission performance before applying the signal to the Mach-Zehnder modulator (MZM). An optical filter was used following the MZM to generate the single sideband optical ODFM signal.

The FPGA functions, which were controlled using a 167.2 MHz clock, are shown in Fig. A bit sequence comprising a 215 DeBruijn pattern and synchronization overhead were stored in a read only memory (ROM) on the FPGA and a block of N = 50 bits was read out each clock cycle. The choice of N depends on the oversampling rate, modulation format and the IFFT size deployed. The size of the IFFT block considered in this work was 128. Oversampling is necessary to separate the OFDM band from the mirror band created by the DAC and was chosen to be 1.28. As a result, 100 out of the 128 subcarriers can be utilized and since digital up-conversion is deployed, 25 channels are used to convey data. This led to an effective bandwidth of 4.18 GHz and a bitrate of 8.36 Gbit/s using QPSK. The 50 bits were modulated to form 25 complex values representing the QPSK constellations and fed to the 26:50 IFFT ports. This maps the OFDM data onto the 4.18–8.36 GHz band. On each clock cycle, the IFFT core generated 128 complex outputs, each with 8 or 10-bit resolution. The choice of IFFT resolutions will be discussed in detail in sections III and IV. In this work, only the real values were used, which were clipped and converted to 4- bit words for the DAC. The clipping circuits operate in an identical way to the scaling circuits used in and the clipping factors (minimum/maximum amplitude and scaling) could be updated during operation. The optimum clipping ratio is dependent on the DAC resolution as described in. The resulting 128 4-bit words were rearranged to 16 32-bit words for output by the MGTs and appropriately delayed for alignment purposes.



(a)

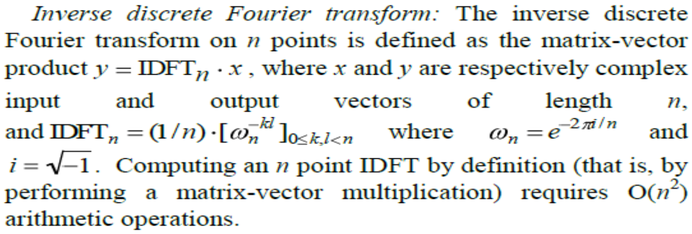
(b)

Fig.1 (a) OFDM transmitter design (b) DSP block functions

III. IFFT CORE DESIGN & ALGORITHM

In this section, we discuss the hardware implementation of the Inverse Fast Fourier Transform (IFFT) used in the OFDM transmitter. In order to meet system performance goals, the IFFT core must be capable of sustaining a very high throughput of one transform (128 points) per cycle. This requirement presents a challenge, but also allows for the consideration of interesting algorithmic options that are not practical in typical hardware implementations of the IFFT.

We utilized an automated tool called Spiral, to explore a wide space of possible options, choose the one that best fits our specific requirements, and automatically generate a hardware implementation of it. Although our current implementation targets a Xilinx Virtex-4 FPGA, our design and methodology are not FPGA-specific; this work can be viewed as a prototype for future ASIC (application-specific integrated circuit) implementation.


Inverse discrete Fourier transform: The inverse discrete Fourier transform on n points is defined as the matrix-vector product $y = \mathrm{IDFT}_n \cdot x$, where x and y are respectively complex input and output vectors of length n, and $\mathrm{IDFT}_n = (1/n) \cdot [\omega_n^{-kl}]_{0 \le k,l < n}$ where $\omega_n = e^{-2\pi i/n}$ and $i = \sqrt{-1}$. Computing an n point IDFT by definition (that is, by performing a matrix-vector multiplication) requires $O(n^2)$ arithmetic operations.

3.1 Inverse fast Fourier transform algorithms*:*

Well-known inverse fast Fourier transform (IFFT) algorithms can be used to calculate an n point IDFT using O(n log n) arithmetic operations. The Kronecker product formalism described in allows an IFFT algorithm to be written as a factorization of the dense IDFT matrix into a product of structured sparse matrices. In general, the IFFT algorithms specified in this way each have the same asymptotic arithmetic cost of O(n log n), but they differ in the degree of regularity exhibited and in the exact number of arithmetic operations.

Spiral is a hardware and software generation and optimization framework that utilizes this mathematical formalism to automatically produce implementations of transforms such as the IDFT. Using Spiral, we explore a wide space of algorithmic and datapath options and automatically generate customized hardware implementations of each.

3.2 IFFT selection and implementation:

Our goal was to implement a hardware core to perform an IFFT of 128 data points on the Xilinx Virtex-4 FX 100 FPGA. The core must operate on fixed-point complex data and must appropriately scale the data to guarantee that overflow will not occur. The implementation must support a throughput of one transform per cycle (128 complex samples per cycle), at a frequency of 167.25 MHz. This gives an overall throughput requirement of 167.25 million transforms per second, or 21.4 billion samples per second.

Typically, hardware implementations of the IFFT require that the IFFT algorithm be very regular, in order to employ *sequential reuse,* where computational elements are time-multiplexed and a small number of data elements are processed at a time. Sequential reuse allows the reduction of circuit area at the cost of slower computation [26]. However, in the proposed application, all 128 data elements must be processed in parallel (as opposed to a few at a time) in order to meet the throughput requirement. So, we must instead implement the IFFT using a *fully unrolled* datapath, which is one that does not include any sequential reuse. Such a datapath will require a large number of computational elements, but will allow additional freedom in the algorithm selection. We evaluated automatically generated IFFT cores derived from two families of IFFT algorithms: one that is very regular and relatively simple, and one that is irregular but with lower computational cost.

3.3 Pease IFFT algorithm:

The radix 2 Pease IFFT algorithm has a highly regular structure and is frequently used in hardware implementations of the IFFT. The algorithm (for *n* data points) consists of log2(*n*) stages, each performing *n*/2 "butterflies," or basic blocks that perform one addition, one subtraction, and one complex multiplication. This algorithm may be viewed as a reordering of the commonly-seen Cooley- Tukey decimation-in-time or decimation-in-frequency IFFT algorithms. Fig. illustrates this algorithm‟s dataflow for *n* = 8; the solid lines show the movement of data between the butterflies. Using Spiral, we generated hardware cores using the Pease IFFT for *n* = 128. Each core requires 2,308 adders/subtractors and 908 multipliers (each operating on real values). Spiral‟s automatic pipelining results in 36 stages. We consider designs with different values of fixed point precision (4, 6, 8, …, 14 bits). For each, we synthesized and place/routed the design using Xilinx Integrated Software Environment (ISE) suite of FPGA tools. In Fig., we plot the number of FPGA slices (reconfigurable logic elements) required for each design (represented as black circles). We observe that only the 4, 6, and 8 bit designs fit on our target FPGA.

Searching across a space of IFFT algorithms: Although algorithms such as Pease are frequently used in the hardware implementation of IFFTs, there are many others one can consider. Typically, these algorithms recursively break down a large IDFT into smaller ones. This type of algorithm can reduce the arithmetic cost, but results in a structure with less regularity.

Using Spiral, we are able to explore many recursive IFFT algorithms and flatten each into an unrolled structure. Then, we can search over many alternatives to find the algorithm with the lowest cost. For this system, the best algorithm found is a mix of radix 8 and radix 16 IFFTs, resulting in 2,192 adders/subtractors and 664 multipliers (a reduction of more than 25% of the multipliers used in the Pease implementation). Again, we use Spiral to generate designs based on this algorithm for various values of fixed point precision. In Fig., we plot the number of slices needed for each design (as black triangles). We observe that designs based on this algorithm require on average 37% fewer slices than their Pease IFFT counterparts. Designs of up to 12 bits of precision are able to fit on the target FPGA, an improvement of four bits.

Numerical accuracy: In Table I, we present an analysis of the root mean squared error (RMSE) and signal to noise ratio (SNR) for the IFFT designs generated from this algorithm. For each design, we perform 512 IFFT computations and calculate RMSE and SNR for each vector. For each design, we report the mean and standard deviation of the 512 values for RMSE and SNR. We calculate RMSE and SNR as follows, where *S* is the measured output vector and *T* is the expected output vector.

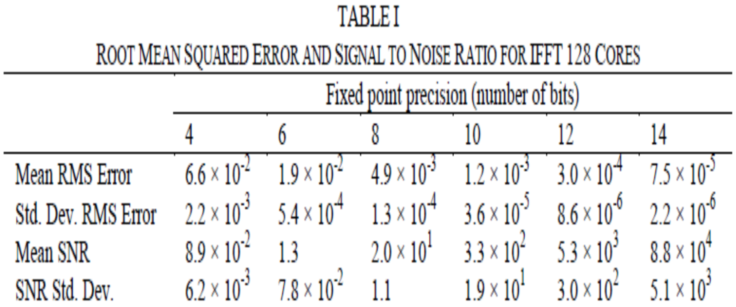


TABLE I
ROOT MEAN SQUARED ERROR AND SIGNAL TO NOISE RATIO FOR IFFT 128 CORES

	Fixed point precision (number of bits)					
	4	6	8	10	12	14
Mean RMS Error	6.6×10^{-2}	1.9×10^{-2}	4.9×10^{-3}	1.2×10^{-3}	3.0×10^{-4}	7.5×10^{-5}
Std. Dev. RMS Error	2.2×10^{-3}	5.4×10^{-4}	1.3×10^{-4}	3.6×10^{-5}	8.6×10^{-6}	2.2×10^{-6}
Mean SNR	8.9×10^{-2}	1.3	2.0×10^{1}	3.3×10^{2}	5.3×10^{3}	8.8×10^{4}
SNR Std. Dev.	6.2×10^{-3}	7.8×10^{-2}	1.1	1.9×10^{1}	3.0×10^{2}	5.1×10^{3}

$$\mathrm{RMSE} = \sqrt{\sum_{n=0}^{N-1} \frac{|S_n - T_n|^2}{N}}$$

$$\mathrm{SNR} = \frac{\sum_{n=0}^{N-1} |T_n|^2}{\sum_{n=0}^{N-1} |S_n - T_n|^2}$$

A 2-bit increase in fixed-point precision causes the maximum magnitude of each data word to increase by a factor of 4. Table I illustrates that such an increase yields a decrease in RMSE of approximately a factor of 4 and an increase in SNR of approximately a factor of 42 = 16.

Pruning: Depending on which subcarriers are utilized, a number of IFFT inputs will always be zero. Because we consider fully unrolled designs, this makes it possible to *prune* out a number of unnecessary operations from the early stages of the IFFT, resulting in a modest decrease in required area. We do this by adding a simple wrapper around the IFFT core and allowing the synthesis tool to perform simplification. In Fig. (white data markers), we evaluate this technique on both IFFT algorithms discussed above, assuming that only a quarter of the IFFT inputs are non-zero. On average, we observe an 8.3% decrease in the number of slices.

IV. SIMULATION MODEL & RESULTS

In this section, we examine the theoretical impact of the limited resolution of the tandem IFFT-DAC on the performance of the transmitter described in section II. The simulation model was setup as follows. First, the hardware description of the DSP section, including the Spiral-generated IFFT cores, was obtained and simulated using ModelSim. The digital output was then fed to a Matlab model that emulated a variable resolution DAC. An ideal DAC with a flat frequency response and negligible electrical noise was considered in the simulation. Finally, an ideal receiver model that includes a true FFT (64-bit floating point) and a noiseless ADC was used to decode and recover the OFDM data.

First, only the analog path was simulated and no optical transmission was considered to isolate the impact of hardware resolution from optical implementation penalties. The quality of the signals was assessed from the resulting constellations through the use of the error vector magnitude (EVM) which is a measure of distances between the ideal constellation and the symbol positions, normalized to the peak constellation symbol magnitude *v*max.

Figure shows the OFDM spectra taken at the output of the DAC and obtained using 8, 10, 12 and 16-bit IFFT with an ideal DAC (no quantization noise). Unwanted frequency components across the 10.7 GHz bandwidth, in addition to the OFDM band, can be observed in Fig. 2 (a) using 8-bit IFFT. These tones are reduced in intensity and occupy a smaller bandwidth (0-3 GHz) when the IFFT resolution is increased to 10-bit (Fig. 2 (b)). The unwanted tones become negligible at 12-bit IFFT (0-1.5 GHz) and disappear when the resolution is increased to 16-bit (Fig 2 (c-d) respectively).

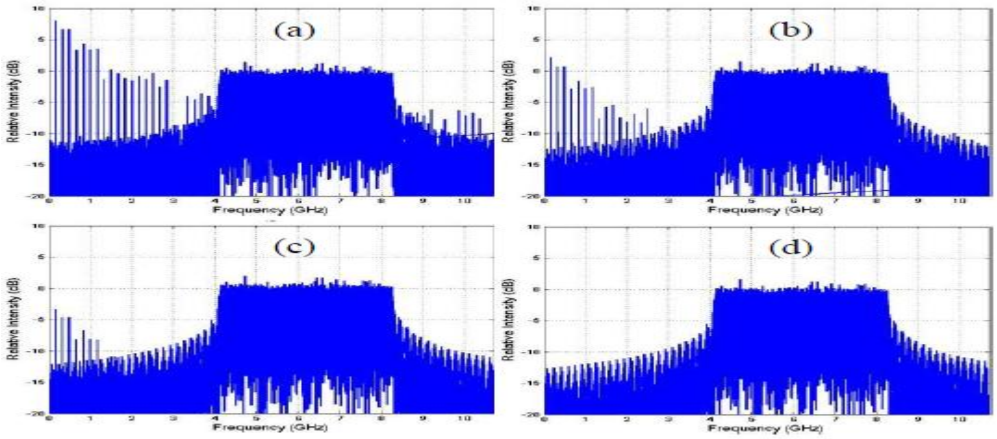


FIG. 2 Simulated electrical OFDM spectra using IFFT resolution of (a) 8-bit (b) 10-bit (c) 12-bit (d) 16-bit

Figure 3 shows the obtained EVM of the system against IFFT precision for 3, 4 and 6-bit DACs. In the case of a 3-bit DAC, an improvement of 1.3 dB in EVM is obtained by increasing the IFFT resolution from 8 to 10-bit. The unwanted frequency components result from the limited resolution of the IFFT core as observed in Fig. 4. This is explained by the fact that at every stage of the IFFT operation, some of the precision (1 bit) is sacrificed after additions/multiplications to avoid system overflow. The loss of precision is more acute for lower resolution arithmetic leading to larger errors and hence more unwanted tones. These tones may interfere with the OFDM data and result in a degradation of the performance. To find out to what extent the IFFT resolution impacts the overall performance of a real system, simulations were carried out using limited IFFT and DAC resolution at the encoder coupled with an ideal decoder.

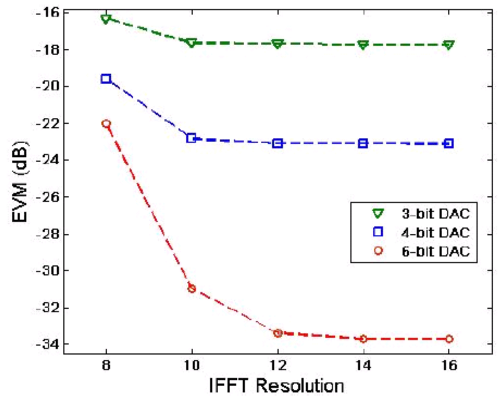


FIG 3. EVM vs. IFFT resolution for 3, 4 and 6-bit DACs

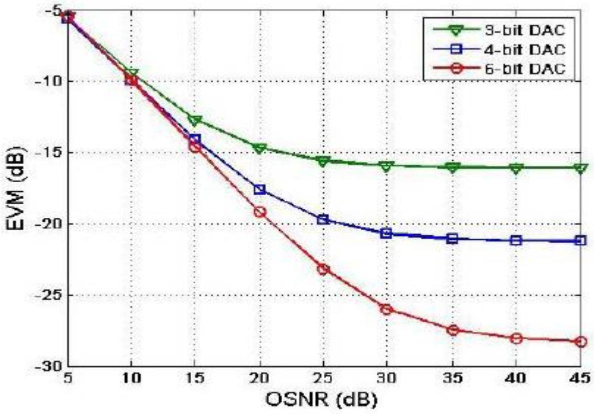


FIG. EVM vs. OSNR for different DAC resolutions

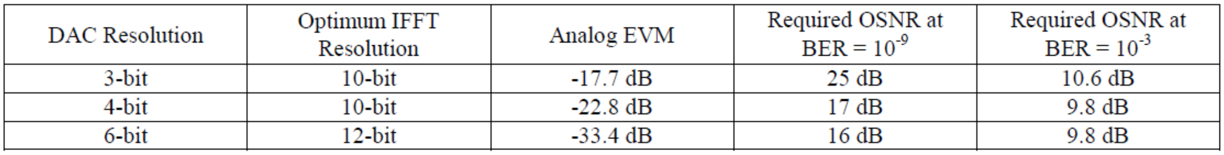
The performance does not improve for higher IFFT precisions and is limited by the DAC resolution. A 3 dB improvement can be observed in the case of a 4-bit DAC by increasing the IFFT resolution from 8 to 10 bits. Increasing precision of the IFFT beyond 10 bits only leads to marginal improvements (0.2 dB). It can be observed in the case of a 6- bit DAC that significant improvements of 9 and 11.4 dB can be achieved by increasing the IFFT resolution from 8 to 10 and 12 bits respectively. Only marginal improvements can be achieved by deploying higher precision IFFT. Therefore it can be deduced that the optimum IFFT resolution for a good tradeoff between precision and complexity is 10-bit for both 3 and 4-bit DACs and 12-bit for 6-bit DACs. Additionally, Fig.3 shows that the overall system performance can be significantly improved by deploying a 6-bit DAC (assuming quantization noise is the dominant source of noise). For instance, increasing the DAC resolution from 4 to 6 bits, improves the EVM of the system by over 10 dB assuming IFFT resolutions of 12 bits or more. After investigating the analog path, the theoretical optical performance is studied next. The transmitter was simulated using an ideal optical filter (brick-wall filter) to suppress one sideband. The receiver consisted of an ideal 11 GHz optical filter, a photodiode, a 10.7 GHz electrical filter and the ideal OFDM decoder previously described. The OSNR (0.1 nm RBW) versus EVM for 3, 4 and 6-bit DACs is shown in above fig. The optimum IFFT precision derived before was used in each case i.e. 10-bit IFFT for the 3 and 4-bit DACs and 12-bit IFFT for the 6-bit DAC.

It can be observed that the required OSNR for a BER of 10-3 (EVM = -9.8 dB) in the case of the 3-bit DAC is 10.6 dB. In the case of 4 and 6-bit DAC resolutions, the required OSNR is 9.8 dB.

V. DISCUSSION

We have seen the impact of the limited IFFT and DAC resolutions on the performance of optical OFDM system in above section. Table 2 is a summary of these results and can be used as a guide for the design of real-time implementations of such systems. The optimum IFFT resolution relates to the best trade off between the output precision and the DSP complexity. The analog EVM represents the quality of the signal at the output of the DAC. These parameters are valid for the proposed transmission design and may change if, for instance, all subcarriers are used to convey data or the IFFT size increases. An *n*-point IFFT consists of log2(*n*) stages of multiplication and therefore an increase in the IFFT size leads to larger errors due to more rounding and truncations. The DAC frequency roll-off also affects the performance of OFDM transmitters which require a relatively flat frequency response. This can be seen in the optical back-to-back performance of Fig. 11, where the low frequency channels are error free while the high frequency ones have a BER of approximately 10-6. Digital pre-equalization of the DAC response is relatively easy to implement and, although results in a slightly lower effective number of bits, will result in an improvement in the system performance.

TABLE 2
DESIGN PARAMETERS FOR DIFFERENT DAC RESOLUTIONS TO GENERATE 8.34GBIT/S SSB DD-OFDM



DAC Resolution	Optimum IFFT Resolution	Analog EVM	Required OSNR at BER = 10^{-9}	Required OSNR at BER = 10^{-3}
3-bit	10-bit	-17.7 dB	25 dB	10.6 dB
4-bit	10-bit	-22.8 dB	17 dB	9.8 dB
6-bit	12-bit	-33.4 dB	16 dB	9.8 dB

VI. CONCLUSION

We have investigated the design of a 21.4 GS/s real-time FPGA-based transmitter used to generate 8.34 Gbit/s digitally-upconverted direct-detection OFDM signals. We have theoretically and experimentally studied the impact of the finite IFFT and DAC resolutions on the performance of the system. The 8.34 Gbit/s signals were generated and transmitted over 800 then 1600 km of uncompensated standard fiber with negligible penalties and BER < 10-3 using a 10-bit IFFT architecture and a 4-bit DAC. We have shown that the transmitter performance can be significantly improved by deploying a 12-bit IFFT core and a 6-bit DAC. This work also shows that FPGAs enable the prototyping and investigation of real-time multi-Gigabit per second OFDM in optical systems.

REFERENCES

[1] J. Armstrong, "OFDM for Optical Communications," J. Lightwave Technol., vol. 27, pp.189-204, 2009.

[2] W. Shieh, Q. Yang, Y. Ma, „107 Gb/s coherent optical OFDM transmission over 1000-km SSMF fiber using orthogonal band multiplexing‟, Opt. Express vol. 16, pp.6378-6386, 2008.

[3] A.J. Lowery, Bangyuan Du Liang, J. Armstrong, „Performance of optical OFDM in ultralong-haul WDM lightwave systems‟, J. Lightwave Technol., vol. 25, pp.131 – 138, 2007.

[4] B.J.C. Schmidt, A.J. Lowery, J. Armstrong, „Experimental demonstrations of electronic dispersion compensation for long-haul transmission using direct-detection optical OFDM‟, J. Lightwave Technol., vol. 26, pp.196 - 203, 2008.

[5] B. J. C. Schmidt, A. J. Lowery, L. B. Du, 'Low sample rate transmitter for direct-detection optical OFDM', in Proc. Optical Fiber Comm.(OFC), paper OWM4, 2009.

[6] W. Shieh, X. Yi, Y. Tang, "Transmission experiment of multi-gigabit coherent optical OFDM systems over 1000km SSMF fibre," Electronics Letters , vol.43, pp.183-184, 2007.

[7] W. Shieh, H. Bao, and Y. Tang, "Coherent optical OFDM: theory and design," Opt. Express vol. 16, pp.841-859, 2008.

[8] H.C. Bao, W. Shieh, "Transmission of Wavelength-Division-Multiplexed Channels With Coherent Optical OFDM", Phot. Technol. Lett., Vol. 19, pp.922 - 924, 2007.