# Fixed-Point Analysis and FPGA Implementation of Deep Neural Network Based Equalizers for High-Speed PON

Noriaki Kaneda, *Senior Member, IEEE*, Chun-Yen Chuang, *Student Member, IEEE*, Ziyi Zhu, *Student Member, IEEE*, Amitkumar Mahadevan, *Member, IEEE*, Bob Farah, Keren Bergman, *Fellow, IEEE*, Doutje Van Veen, and Vincent Houtsma

***Abstract*—A deep neural network based equalizer is proposed to mitigate the intersymbol interference observed in next generation high speed passive optical network (PON) links. The DNN based equalizer is shown to outperform the best known conventional equalizer, the maximum likelihood sequence estimator (MLSE) both in back-back and through fiber experiments. To reduce the hardware complexity of DNN based equalizer for PON systems, we investigate the use of embedded parallelization within a DNN structure having multiple symbol outputs from one DNN. We further investigate using a classification output stage with cross entropy cost to perform joint decision on multiple symbol outputs and demonstrated that the sensitivity gain of such scheme over regression output. To understand the complexity of hardware implementation, the fixed-point DNN based equalizers are developed and implemented in FPGA. The impact of fixed-point resolution on the receiver sensitivity and hardware resource utilization in FPGA implementation is analyzed and reported in detail. We show that a reduction of over 40% in LUTs (look up table) utilization is possible by reducing the DNN's weight resolution from 8-bit to 4-bit while incurring a small penalty in receiver sensitivity.**

***Index Terms*—Artificial intelligence, feedforward neural networks, neural network hardware, neural networks, optical fiber communication.**

## I. Introduction

While the highest speed PON system available today is at 25Gp/s as per 25GS-PON MSA (DNN) (multi-source agreement) [1] and IEEE 802.3ca [2], the ITU-T has already approved the next generation PON standard (G.hsp) with 50Gbaud NRZ downstream signaling [3]. One of the most distinguishing aspects of the G.hsp PMD (physical medium dependent) layer is that for the first time in a PON standard, the received signal is expected to be equalized using DSP (digital signal processing). The necessity of such signal equalization arises from the fact that 50Gbaud signal experiences intersymbol interference (ISI) both from the signal bandwidth limitation of cost-effective components and chromatic fiber dispersion at 1342 +/−2 nm wavelength designated by the standard. The current assumption for DSP in G.hsp is based on the use of FFE (feedforward equalizer) which achieves an optical path penalty (OPP) of 3.5 dB after equalization for 20 km of worst case fiber at the 1344 nm wavelength. Even though such traditional DSP is applicable to recover large part of the penalty caused by the signal distortion, ISI mitigation in IMDD (intensity-modulation direct detection) system is worth the careful attention for potentially improving power budget due to reduction in OPP and enabling longer reaches by being more tolerant to ISI. This is because the ISI in IMDD is generally not an 100% reversible effect in which there is signal loss in high frequency components due to bandwidth limitation and signal fading due to transmitter chirp and fiber's chromatic dispersion. The choice of the equalizer and its effectiveness can have a non-negligible impact on the receiver sensitivity and the optical path penalty, both of which are important in PON to meet the stringent power budgets as well as reach.

Although machine learning for signal equalization was proposed as early as in 1990 [4], significant increase in research interest of machine learning as signal equalizer is seen only recently [5]–[8]. The deep neural network (DNN) as one of the machine learning techniques is the general term which refers to multilayer neural networks with no specific topologies of how neural networks are connected. Among many possible DNN architectures, the most conventional feedforward deep neural network, which is the same as multilayer perceptrons (MLPs) [9], are extensively studied for signal equalization techniques in highspeed optical communications [8]. The feedforward DNN has the information flow from the input to output of neural networks without feedback path and is analogous to the finite impulse response (FIR) filter [10] in conventional DSP design.

Manuscript received July 7, 2021; revised September 25, 2021 and November 9, 2021; accepted December 2, 2021. Date of publication December 10, 2021; date of current version April 4, 2022. *(Corresponding authors: Amitkumar Mahadevan; Noriaki Kaneda.)*

Noriaki Kaneda was with Nokia Bell Labs, Murray Hill, NJ 07974 USA. He is now with the nEYE systems, Berkeley, CA 94710 USA (e-mail: nkaneda@ieee.org).

Chun-Yen Chuang was with Nokia Bell Labs, Murray Hill, NJ 07974 USA. He is now with the Department of Photonics and Institute of Electro-Optical Engineering, National Chiao-Tung University, Hsinchu 30010, Taiwan (e-mail: ryoma3016.eo97@g2.nctu.edu.tw).

Ziyi Zhu was with Nokia Bell Labs, Murray Hill, NJ 07974 USA. He is now with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: zz2374@columbia.edu).

Amitkumar Mahadevan, Bob Farah, Doutje Van Veen, and Vincent Houtsma are with Nokia Bell Labs, Murray Hill, NJ 07974 USA (e-mail: amitkumar.mahadevan@nokia-bell-labs.com; bob.farah.ext@nokia.com; dora.van_veen@alcatel-lucent.com; vincent.houtsma@nokia-bell-labs.com).

Keren Bergman is with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: bergman@ee.columbia.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JLT.2021.3133723.

Digital Object Identifier 10.1109/JLT.2021.3133723

This is the most practical and feasible architecture of DNN suitable for the highspeed communication with over Gbaud signal throughput. In addition to dealing with the high throughput in optical communications, the application space of DNN based techniques may be limited to generally static channels rather than rapidly changing dynamic channels. This is because the adapting speed of the DNN neurons are generally limited due to the sheer number of neuron updates required to achieve convergence in DNN training. The DNN as signal equalizer for PON or short reach optical link is attractive as the optical link is based on IMDD principle where the source of the signal distortion in the received signal is mostly static and deterministic after successful symbol timing recovery [11]. In the previous works [5]–[8], the effectiveness of DNN as signal equalizer is reported and compared to various equalizers. Based on those reported results [8], it is safe to conclude that the DNN based equalizer can be as good as or even better than the conventional equalizers such as FFE, FFE/DFE (decision feedback equalizer) and MLSE (Maximum Likelihood Sequence Estimation) [12], [13].

Since the effectiveness of DNN as equalizer has been validated, the next logical important question is how much hardware resources are required to implement DNN. Although previous research on a physical layer DSP implemented in a general-purpose GPU (graphic processor unit) has been reported [14], such a solution is generally not preferred for a choice of physical layer DSP hardware in high speed optical communications transceivers operating at tens of Gbaud due to the higher cost and higher power consumption of a GPU compared to an ASIC (application specific integrated circuit). Further, prototyping ASIC implementations on FPGA is a well-established methodology for rapid evaluation of design choices and implementation complexity. While many studies have been reported on FPGA prototyping of various DSP techniques in optical communications [15]–[20], most of the work published so far for the DNN applications in optical communications are using offline processing [5], [7]. Further, hardware implementation of DNN for optical applications that do not demand very high throughput have been investigated in [21], [22]. A much higher throughput is required when DNNs are used for physical layer equalization [6], [8], [23]. Our previous work [8] and, also in [23] have taken initiatives in studying the requirement of hardware resource when DNN is used as signal equalizer in an 50G PON system by implementing DNNs in a FPGA. The work in [6] however uses implementations with 8-bit fixed-point resolution and no other fixed-point resolutions were considered while the FPGA implementation was utilizing DSP resources that can implement MAC (multiplier and add computation) with dedicated hardware which obscures true hardware complexity of the implemented DNN. In addition, DNNs with multiple outputs are commonly considered in the machine learning community [9] and have also been proposed as a part of DNN application in the physical layer equalization [24] for general communication systems. However in optical communities including PON systems, where one of the highest data throughput is demanded among all communication systems, an extensive study on the parallel output DNN equalizer has not been considered until it was reported in [8]. In this work,

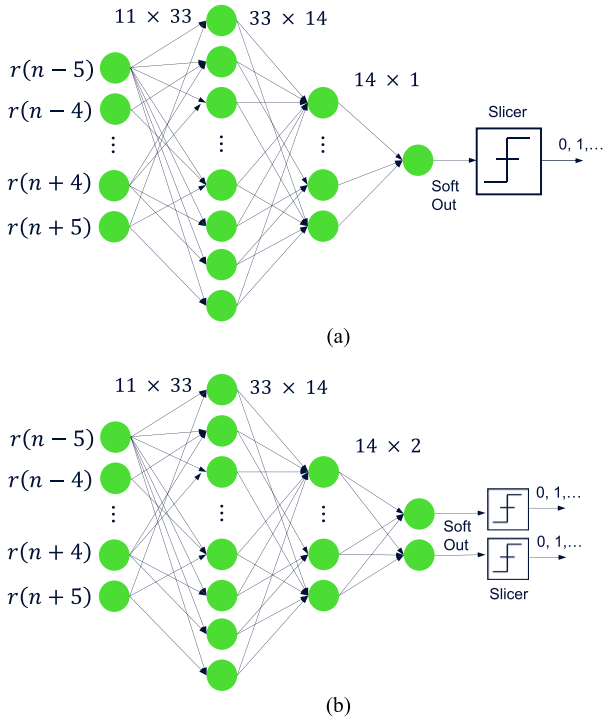



Fig. 1. Deep neural network with 2 hidden layers for NRZ signal equalization. (a) with 1 linear soft symbol output (b) with 2 linear soft symbol outputs.

we analyze the impact of reducing the fixed-point resolution on the neural network performance which is evaluated in terms of receiver sensitivity curve using experimental data obtained from a 50G PON link. We then demonstrate how the reduced resolution DNN can lower the hardware resource needed by implementing designs with various fixed-point resolution in FPGA. We also investigate a neural network with joint classification based output stages in embedded parallel DNN architecture as opposed to the regression-based (linear) output stage computation. The obtained results for this configuration suggest such classification output can improve DNN equalizer performance especially when the resolution becomes very low.

## II. Deep Neural Network With Multiple Outputs

### *A. Embedded Parallelism With Linear or Classification Outputs*

We choose the standard feedforward DNN architecture as shown in Fig. 1 to construct a signal equalizer for a distorted NRZ signal from 50Gbaud G.hsp PON link. The structure has no feedback path so it is suitable for pipelined implementation and therefore can achieve high data throughput. The DNN shown in Fig. 1 has 11 inputs sampled consecutively at symbol rate, 2 hidden layers with 33 and 14 neurons, and one or two soft outputs before the output is sliced for 0 or 1 output. The number of outputs can be either one in Fig. 1(a) or two as in Fig. 1(b) with each representing a soft output of one or two OOK symbols, respectively. In this way, parallelization is embedded within the DNN design to help reduce the hardware resources in the implementation. The neurons between layers are fully connected

as follows,

$$\vec{a}_k = \sigma\left(\boldsymbol{W_k}\vec{a}_{k-1} + \vec{b}_k\right) \tag{1}$$

$$\sigma(z) = \begin{cases} 1 - 2^{1-D}, \, when \, z > z_{max}, \\ z, \, when \, z_{max} \geq z \geq 0, \\ 0, \, when \, z < 0 \end{cases} \tag{2}$$

$$\sigma(z) = \begin{cases} 1 - 2^{1-D}, \, when \, z > z_{max}, \\ z, \, when \, z_{max} \geq z \geq z_{min}, \\ -1, \, when \, z < z_{min} \end{cases} \tag{3}$$

Equation. (1) is representing the input-output signals of each layer where $\overrightarrow{a_k}$ is the output signal vector of each layer, $\overrightarrow{b_k}$ is the bias vector and $\boldsymbol{W_k}$ is the weight matrix at $k$th layer. We use the data range between $-1$ to $1-2^{1-D}$ to capture the constraints of fixed-point representation where $D$ is the number of fixed-point bits. Therefore, the fixed-point quantized and saturated version of the rectified linear unit (ReLU) which is represented in (2) is used as the activation function $\sigma$ for the 2 hidden layers, while the fixed-point quantized and saturated linear function shown in (3) is used at the output neurons. Note that both (1) and (2) have bounds at $1-2^{1-D}$ or $-1$ which may not be normally considered in floating-point computations, but this is necessary for fixed-point computations. While the usage of bipolar activation functions [6], [7] have been reported previously, ReLU is unsigned with 1 bit more efficient usage of resolution and the simplest to realize in hardware. The output layer in the Fig. 1 has linear regression output and the soft output that are sliced at zero to provide binary symbols. The cost function for DNN training is mean square error between the true symbol and the estimated soft output. The same cost function and the same procedure of training are used for both 1 output and 2 output DNN.

In Fig. 2, on the other hand, we show basically the same DNN as Fig. 1 except (a) with a classification output layer with one output symbol and (b) with joint classification output layer with 2 simultaneous symbol outputs. When we use classification output, we double the number of output neurons which does the same linear regression operation as defined in (1) with linear activation function in (3). The difference is that the output from one output neuron in Fig. 2(a) represents the probability of the symbol interpreted as logic 0 and the other output neuron represents the probability of the symbol interpreted as logic 1. In Fig. 2(b), we propose to use 4 output neurons with linear output (logit) as in (3) just as the same way as Fig. 1. To train the neurons, we use softmax function [9] defined in (4) as our probability function and minimize the cross entropy between the target probability distribution and the tested probability distribution [9]. The goal is to maximize the probability of the logit to be one of the four classes denoted as $i = \{1, 2, 3, 4\}$ in (4). The classes correspond to the four possible consecutive bit patterns, and they are 00, 01, 10 and 11.

$$\mathrm{P}\left(z_i = i \,|\vec{x}_k\right) = softmax\left(\vec{z}\right) = \frac{exp\left(z_i\right)}{\sum_j exp\left(z_j\right)} \tag{4}$$

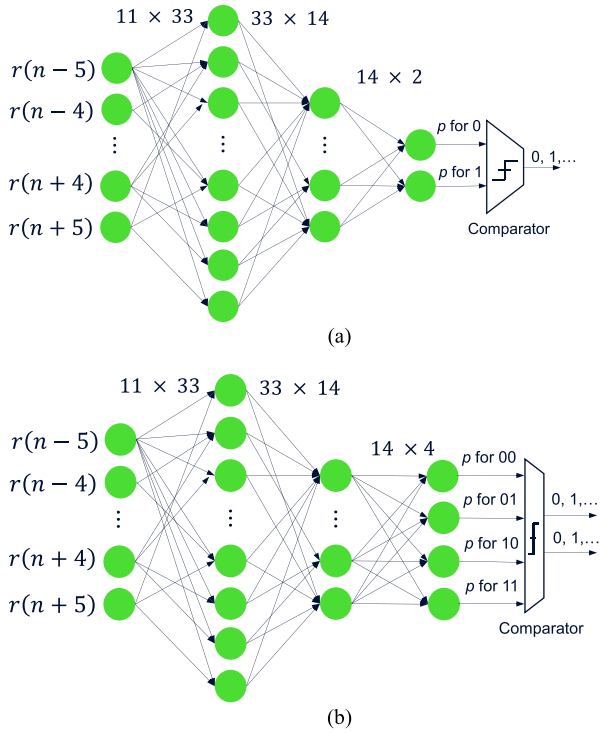



Fig. 2. Deep neural network with 2 hidden layers for NRZ signal equalization. (a) with a classification output stage for 1 output symbol (b) with joint classification output stage for 2 output symbols.

Since we are dealing with 2 consecutive output patterns in one classification training rather than dealing with two outputs independently, we consider this a joint classification output stage.

### B. Hardware Efficiency Improvement of Parallelized DNN

When we consider the hardware complexity of DNN based equalizers, we can obtain good insights by comparing it to a conventional FIR filter implementation. One of the major advantages of conventional FIR filter is its adaptability to frequency domain implementation with overlap-save (OLS) approach [10] for hardware efficient parallel data processing. Although the feedforward DNN is constructed similarly to a FIR filter, due to its nonlinear function in activation layer as well as its fully connected neuron topologies, it is not straightforward to find efficient frequency domain implementation of DNN. Instead of finding frequency domain translation of DNN, we can consider DNN to directly perform overlap-save filter with both time-domain inputs and outputs data stream. In this view, the major difference between DNN based overlap-save and the conventional overlap-save filter implementation is the overlap rate [25] which can be defined using the number of input samples (L) and the number of output (N) as $R = N/L$. In this definition, the overlap ratio corresponds to data throughput of the digital filter. Although the required overlap rate of the overlap-save filter depends on the impulse response of the filter, using DFT (discrete Fourier transform) as linear convolution within overlap-save scheme, the smallest overlap rate for conventional FIR filter implementation

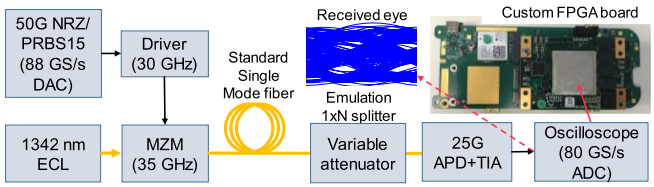



Fig. 3. Experimental setup of 50Gb/s downstream PON at 1342 nm. The captured data in the scope is then transferred to FPGA board for on chip processing.

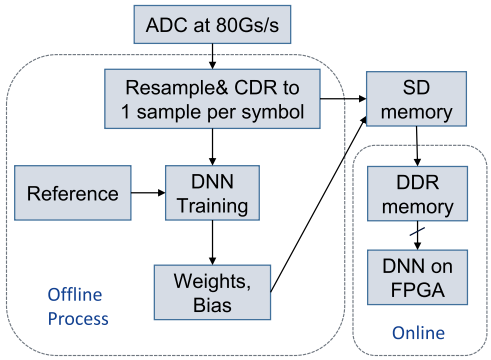



Fig. 4. Design flow of DNN hardware implementation. The captured data based on experimental setup is used for training DNN in offline process. While the DNN inference is implemented in FPGA with pipelined and continuous data processing.

is 50%. The overlap rate of the DNN filter on the other hand can be very low. For the case of Fig. 1(a) which has 11 input and 1 output, the overlap rate is 9.1%. For Fig. 1(b) the overlap ratio is 18.2% which is still much smaller compared to the conventional FIR filter implementation but at least it is doubled compared to the Fig. 1(a) DNN.

In the case of linear system with OLS, we know that the requirement of overlap comes directly from the impulse response of the filter, in this case the length of the inverse channel response to deal with the dispersive channel. For DNN based equalizer, the same analogy should apply where the input size needs to be increased when the output size is increased. In addition, we observe the number of output symbols can be more than 1 without significantly sacrificing BER performance only when the first hidden layer is sufficiently larger than the input width. This observation is consistent with the study reported in [26] of 2 hidden layer DNN being sufficient.

The weights and biases are obtained by the backward propagation method of DNN using Python TensorFlow with quantization embedded into the outputs, weights and biases of all DNN layers during the training. Note that the training of DNN especially for downstream PON link can be done offline even in the field deployment, since the IMDD channels are essentially stationary and exhibit slow changes only primarily due to temperature fluctuations.

## III. Experimental Setup and FPGA Implementation

Although FPGAs provide useful prototyping platforms, an FPGA has limited hardware resources and IO data throughputs. Due to such practical limitations, some of the past DNN demonstrations in FPGAs are performed at low signal baud rates at around 4Gbaud [6]. Low signal baud rate signals, however, do not face similar ISI issues that high baud rate signals experience from chromatic dispersion with practical transmission distance, therefore the equalizer capability and requirement are more difficult to evaluate for 50G PON systems with such low level of ISI. It is therefore important that equalizer complexity in hardware is evaluated with practical channel distortion at practical signal speeds. In this work, we use 50Gbaud signal baud rate and an optical link specified in G.hsp as the target optical link where the ISI from components bandwidth limitation and chromatic dispersion is a real issue with as much as 3.5 dB optical path penalty [3] even after equalization for an FFE type receiver equalizer.

The experimental setup of 50 Gb/s PON is shown in Fig. 3. The Lithium niobate (LN) Mach-Zehnder modulator (MZM) is used for ease of downstream wavelength selection. We set the tunable laser wavelength to 1342 nm to emulate the elevated amount of dispersion for PON downstream. We used up to 30 km single-mode fiber to test the capability of our DNN equalizer. The zero dispersion of the fiber used is believed to be ∼1310 nm, translating into 83 ps/nm total fiber dispersion for 30 km which is equivalent to 20 km of fiber with worst case zero dispersion wavelength per G.hsp [3]. The signal is modulated with 88 GS/s CMOS 8-bit DAC (digital to analog converter) with pulse shaping to compensate for the bandwidth profile of the DAC and its eval board. The exact data rate used is 50.2857 Gb/s to adopt 1.75 sample per symbol. The 25 Gb/s class APD (avalanche photo diode) integrated with TIA (transimpedance amplifier) is used as the photoreceiver and the output of the receiver is captured in the real-time sampling scope. The received eye in Fig. 3 shows large distortion from bandwidth limitation and ∼83 ps/nm fiber dispersion. The same captured data is used to compare DNN based equalizer and MLSE which is one of the best conventional equalizers.

Fig. 4 illustrates the design flow of the DNN implementation in FPGA. The captured data is first preprocessed offline down to 1 sample per symbol and used in the offline DNN training, verification and testing offline. Each measured dataset has about 1e6 time-domain samples in 8-bit effective resolution and the first 2.5e5 samples are used for training, the next 1e5 samples for verification and the following 5e5 samples for testing. The training of the DNN is done separately for each measured point and for each DNN. The number of epochs used is typically 50. The training and testing results are found to be quite similar and the overfitting problem is avoided by careful design of the choice of pseudorandom data pattern and the input size of DNNs as described later. The same measured data are also later used for testing of the DNN in FPGA. The FPGA used in the test is the Xilinx XCZU9EG-FFVC900 device which is a ZynqMPSoC device with 4 onchip ARM application processors [27]. The custom FPGA carrier board is developed to interface with a Trenz Electronic SoM (system on module) child board that has 4 GByte DDR4 memory directly connected to the PS

(processor system) side of the FPGA. The measured data as well as the weights and bias values are stored first in a SD (secure digital) card, then read into the DDR4 memory and finally are loaded into the FPGA fabric for the online DNN process. Using Petalinux tools [28], we build the embedded Linux kernel on Xilinx FPGA, and use it as the operating system (OS) to handle the data flow between storage, memories and FPGA fabric as well as to interface the user commands. We use a PRBS15 pattern based on the polynomial $X^{15}+X^{14}+1 = 0$ as transmit pattern. Since we center the 1 or 2 output symbols in the middle of the DNN input signal vector as in Fig. 1, the DNN is not capable of learning the PRBS15 pattern until DNN input width exceeds 28 [29], therefore it is ensured that the presented results are not overfitted. To verify that DNN is learning channel distortion but is not overfitted to the noise (mostly thermal in our case), we cross check the obtained weights and biases from one input power with other input power levels of the same channel and DNN conditions, and ensure the results are approximately the same. Furthermore, the sensitivity values obtained after DNN equalization in this measurement come close but does not exceed the expected 50G sensitivity derived from (non-equalized) 25Gb/s sensitivity [30] while overfitted results often exhibit the performance that are even better than theoretical values by many decades in BER [29].

For preparation of RTL (register transfer level) description of DNN for FPGA implementation, we use Xilinx Vivado high level synthesis (HLS) [31] in this work. The HLS can semi-automatically translate C programming codes to hardware description language (HDL) codes and has been gaining both capability and popularity over the years due to the fast development cycle for the initial prototyping. One of the key aspects in writing hardware in C language is the throughput of the logic we implement. This is because HLS by default produce the RTL logic that requires multiple clock intervals to generate the output, and this can lead to the designed circuit that is not capable of handling signal streams. To address this problem, we carefully pipeline all steps of the DNN computation represented in (1)-(3) including matrix element multiplications and summation at each layer to make sure that the throughput is 100% of the data rate and the logic is capable to process the signal stream at every clock cycle.

## IV. DNN Equalizer Performance and FPGA Hardware Resource

### A. DNN Equalizer in Comparison to Conventional Equalizers

The DSP diagrams for processing captured data both in MLSE and DNN in Fig. 1 are shown in Fig. 5. The samples at 80Gs/s are first upconverted to approximately 2 times oversample data at 100.5714 Gb/s before the clock recovery and decimation to symbol spaced samples are applied using the technique explained in [11]. The BER vs received optical power for both MLSE and 8-bit fixed-point DNN with a regression output stage as in Fig. 1(a) and (b) are plotted in Fig. 6. The number of taps used for MLSE is 6 and no gains are observed by increasing MLSE taps beyond 6 for both BtB (back-to-back) and 30 km data. When they are compared at BER = 1e-2, the DNN outperforms

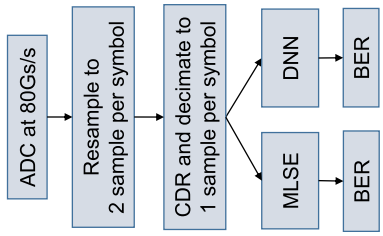



Fig. 5. DSP diagrams for MLSE and DNN processing. The data captured at 80Gs/s is first symbol synchronized and is decimated to 1 sample per symbol.

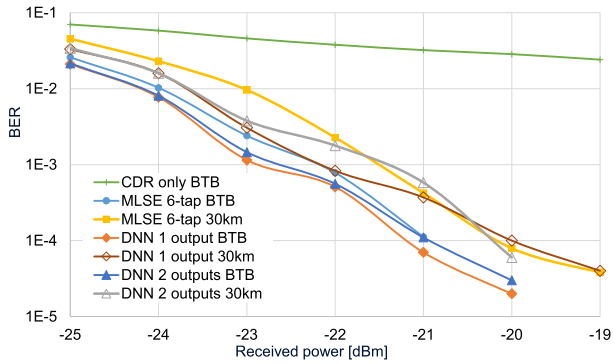



Fig. 6. Measured sensitivity of MLSE and DNN receiver equalizers for 1342 nm 50Gb/s PON link. sample per symbol.

MLSE by 0.7 dB in sensitivity with 30 km link and ~0.2 dB with back-back measurement. The DNN results for 1 and 2 symbol outputs are almost overlapping near 1e-2 while the 1 output DNN outperforms at BER = 1e-3 by 0.6 dB. This indicates that we can find a DNN topology with embedded parallelization which results in small performance penalty. However, when the number of output symbols of a single DNN are increased beyond 2, the presented DNN shows a larger BER degradation. Note that we show the BER results with more than 5 error counts per data point with fixed sample size, as such the confidence level near 1e-5 BER point is lower.

### B. FPGA Resource of Highest Throughput DNN At 8-bit Resolution

There are two parts in the study of required FPGA resources needed for DNN based equalizations for 50G PON links. We will first present the hardware resource needed to achieve the highest data throughput by using an 8-bit fixed-point resolution for both 1 output and 2 output DNN before moving on to studying the impact of reducing the fixed-point resolution on DNN performance as equalizer and the impact of it on hardware resources. Table I shows the summary of the resource utilization for 2 different DNN designs implemented in FPGA. One has 1 symbol output while the other has 2 symbol outputs per DNN as shown in Fig. 1. For both DNN designs, 4 copies of DNNs are implemented to increase the total throughput. The resource utilization shows only a small increase for the 2 outputs DNN while its throughput is doubled. All designs met the timing at 325 MHz FPGA clock with 100% throughput, with a total data throughput up to 2.6 Gb/s is presented in Table I. To meet the

TABLE I
RESOURCE UTILIZATION OF 2 DIFFERENT DNN EQUALIZERS IMPLEMENTED IN A FPGA

| DNN equalizers | 11,33,14,2 | 11,33,14,1 |
|---|---|---|
| *Outputs per DNN* | 2 | 1 |
| *Parallel DNNs* | 4 | 4 |
| *LUT, LUTRAM* | 74%, 20% | 74%, 19% |
| *FF, DSP48* | 31%, 100% | 29%, 100% |
| *Data throughput* | 2.6 Gb/s | 1.3 Gb/s |

LUT: look-up table, RAM: random access memory, FF: Flip-flop, DSP48: 48-bit DSP unit.

50Gb/s data rate, the same DNN needs to be parallelized by 20 times. This suggests higher resource requirements for DNN equalizer compared to the 32-state MLSE (equivalent to 6-tap) implemented in a similar-size FPGA for 10 Gb/s [13] albeit an exact comparison is not available. To further reduce hardware complexity a number of approaches including reducing the total number of neurons, pruning neuron cross-connects, having more parallel outputs per DNN need to be explored.

### C. DNN Performance Impact With Reduced Resolution

It is expected that DNN performance is impacted when the fixed-point resolution is reduced but it is important to quantitatively understand the degradation of equalizers' performance. In this section, we present results on how the reduced resolution impact DNN performance as equalizer and if different DNN structure as shown in Figs. 1 and 2 have any impacts. Fig. 7 shows the measured receiver sensitivity for both BtB and 30 km link experiments with 11x33x14x1 DNN architecture, when the fixed-point resolution is reduced from floating point (16-bit) all the way down to 2-bit. Since the hardware resources are expected to be mainly constrained by the multipliers in the DNN equations, we decided to only change the resolution of the DNN weights and kept all other signals namely input, output, and bias at each layer to 8-bit resolution. For our previous results as depicted in Fig. 6 and also reported in [8], an 8-bit resolution was used for all signals including weights. From Fig. 7 it can be observed that the resolution of weights can be reduced to at least 4-bit without any significant impact on performance, namely degradation on BER, for the DNN architecture of 1 output. When the resolution of the weights is reduced further to 3-bit, however, we see about 0.5 dB sensitivity penalty at BER = 1e-2 and at 2-bit resolution we see about 1 dB penalty for BtB and approximately 2 dB penalty for a 30 km fiber link. The DNN with classification output stage shows slightly better performance but the most improvement in this case is still limited to negligible value ~0.2 dB with a 30 km fiber and at very low resolution.

Fig. 8 shows the measured receiver sensitivity for both BtB and 30 km link experiments with 11x33x14x2 DNN architecture, when the fixed-point resolution of the weights is reduced from floating point (16-bit) down to 2-bit. The main difference

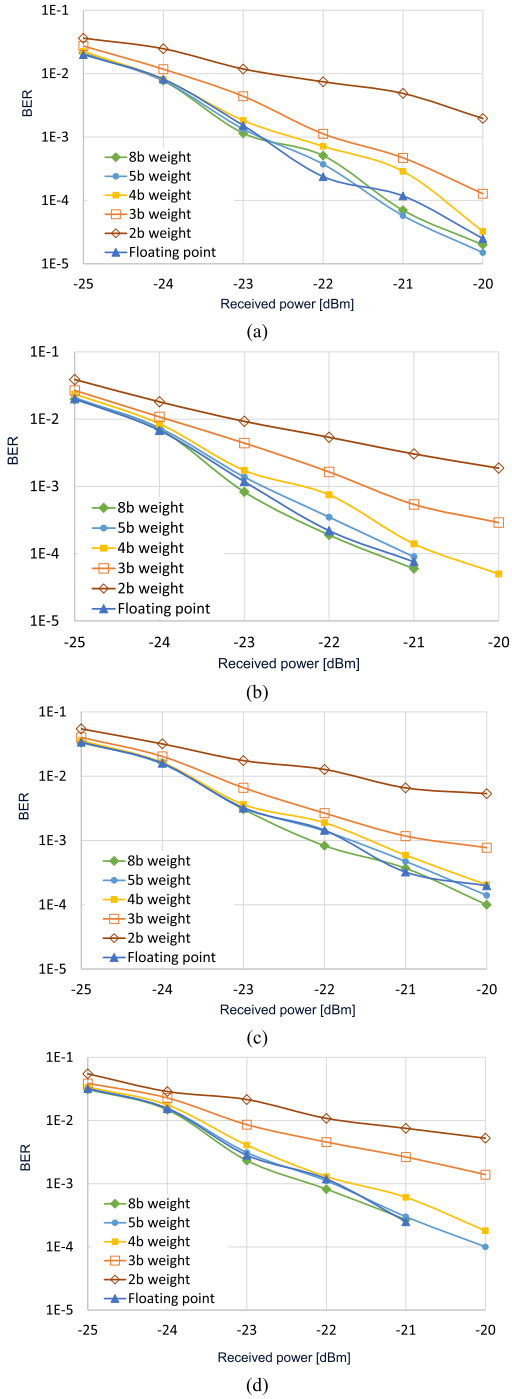
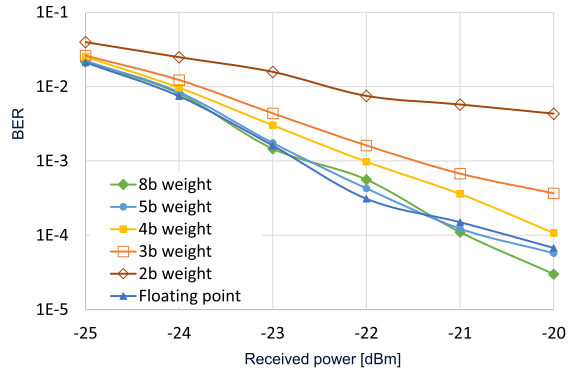
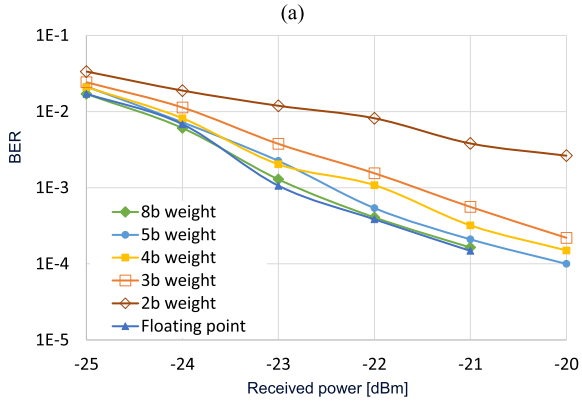



Fig. 7. Measured sensitivity of DNN receiver equalizers based on 11x33x14x1 architecture with various fixed-point resolutions. (a) For BtB link with fixed-point linear output stage (b) for BtB link with classification output stage (c) for 30 km link with linear output stage (d) for 30 km link with classification output stage.

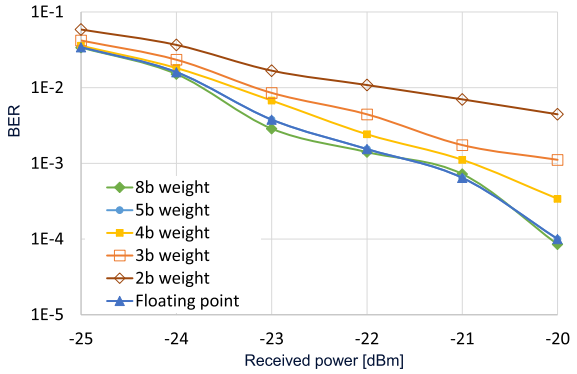
compared to Fig. 7 which shows the result with 1 output DNN is that the performance of the DNN already starts to degrade when the resolution is reduced to 4-bit. Especially when there is 30 km fiber, the result with 4-bit weight resolution shows about 0.3 dB penalty at BER = 1e-2 when two parallel linear output stage is used. This penalty however is entirely removed



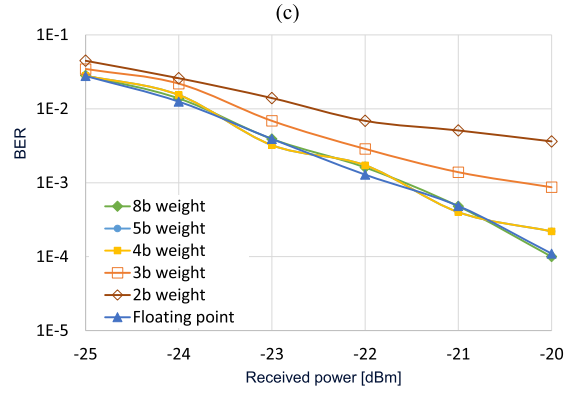

(a)




(b)




(c)




(d)

Fig. 8. Measured sensitivity of DNN receiver equalizers based on 11x33x14x2 architecture with various fixed-point resolutions. (a) for BtB link with linear output stage (b) for BtB link with classification output stage (c) for 30 km link with linear output stage (d) for 30 km link with classification output stage.

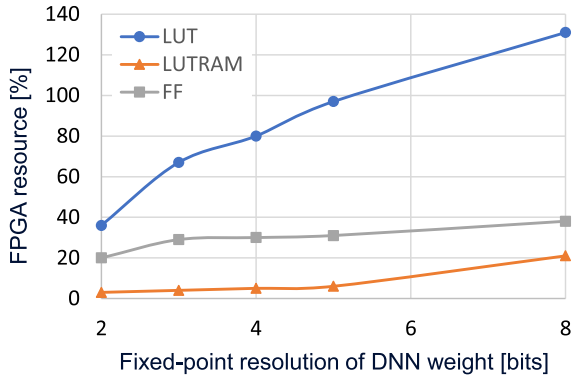



Fig. 9. FPGA resource utilization vs. fixed-point resolution of DNN weight. The estimated resource is after synthesis not implementation.

when a joint classification output is used as in Fig. 2(b). Also, when the resolution of the weights is further reduced to 3-bit and even 2-bit, the joint classification output outperforms the linear parallel output with small improvement in receiver sensitivity. Based on these results in Figs. 7 and 8, we can conclude that even though a joint classification output stage does not improve the DNN performance when the hardware's fixed-point resolution is sufficiently high, it helps to mitigate the quantization distortion when the hardware resolution is reduced to 4-bit or lower.

Once we obtain the RTLs based on the procedure described in the previous section, the FPGA hardware resource is analyzed in Xilinx Vivado tool [32] with the target implementation in Xilinx XCZU9EG-FFVC900 FPGA device. As opposed to the previous implementation, since this effort is specifically targeted to understand the hardware resource utilizations of different fixed-point resolution, we implement all designs using FPGA standard logic resource namely LUT (look-up table), LUTRAM (look-up table random access memory) and FF (flip-flop) without using DSP blocks in the device. The results are shown in Fig. 9 which corresponds to the implementation of DNN architecture depicted in Fig. 1(b) with various fixed-point resolution. The hardware resource utilization numbers are coming from the post-synthesis reports not post-implementation reports. This is because when 8-bit resolution is used, the post-synthesis hardware utilization exceeds 100% of available LUT and no longer fits in a single FPGA. Despite the resource utilization exceeding 100%, post-synthesis hardware utilization is generally considered to be valid and accurate. This is verified in the designs with up to 4-bit resolution where the designs do fit in a device and the post-implementation utilization are basically the same (within 1% difference) as the post-synthesis utilization estimate. The results show how the resolution change of DNN weights alone proportionally affects the hardware complexity of the entire DNNs. One of the important facts we can observe from the plot is that the LUT resources are reduced by almost 50%, by going from 8-bit to 3-bit resolution on DNN weight. In Fig. 9, the increase in FF utilization is not observed as the resolution for the weight increases. This may indicate that the multiplication logics in these FPGA designs are mostly realized in LUTs with

little help from FFs. At the pipeline stages however between multipliers and DNN layers, we always use 8-bit resolution regardless of weight resolution. Based on Fig. 9, we believe FFs are mainly used for the pipeline registers but not in multiplier logics.

The general complexity of DNN is roughly proportional to the number of real multiplications between weights and neuron inputs. For the fully connected DNN studied in this work, the number of multipliers can be computed as $\sum_{k=2}^{L} N_n(k) \cdot N_n(k-1)$ where $N_n(k)$ stands for the number of neurons at the $k$ stage of the DNN with total $L$ stages. For the Figs. 1(a) and 2(a) the number of real multiplications is 839 and Figs. 1(b) and 2(b) the number of multiplications is 853. By varying the resolution of the weights involved in the multiplication, we can learn the impact of the complexity of each multiplication on both BER performance and FPGA resource utilization. We can use the scaling based on the number of multiplications to apply the results presented here to different DNNs. The presented results here therefore can be used as a guideline to design and to implement a DNN with a good balance between DNN performance and DNN power consumption and/or throughput. In addition, we can view these results as further study points for the trade-off between resolution and the size of DNN.

## V. DNN Equalizer for Upstream PON Link

PON is a point to multi-points system and while the downstream link uses continuous signal streams, the upstream link is operated with burst mode data streams. Although we have focused on DNN equalizer as a downstream receiver equalizer, all our analysis on the hardware complexity of DNN inferences is applicable regardless of whether DNN is used in downstream or upstream links. The additional technical requirements for upstream link however need to be addressed to cope with the multiple end users with different channel responses. Although such study is out of scope in this work, there are a few different approaches that are related to our study. One is to use over-dimensioned layers so that a DNN can be trained to cope with multiple users multiple channel responses [7]. The other technique [33] also relies on over-dimensioned DNN to cope with multi users channel response but with help of a dedicated part of NN (sub-NN) with received signals amplitude histogram as inputs. In either technique, the fully connected DNN is the basis of DNN equalization, and our study of hardware complexity should serve as a valuable reference point in their hardware implementations. The other approach to deal with the upstream traffic can be based on the fact that PON's OLT (optical line terminal) has full knowledge of ONU (optical network unit) schedule. With such knowledge at OLT, once the DNN is trained for different ONUs, we can swap out the DNN weight and bias in time for the arrival of each ONU data frame. The latter approach can let us avoid using over-dimensioned DNNs which have higher hardware complexity than the optimally designed DNNs [7].

## VI. Conclusion

While the performance of DNN based equalizers has been demonstrated to exceed conventional equalizer capability, its hardware complexity, and the research into approaches needed to reduce hardware complexity has just started. This work extends our previous work [8] of DNN equalizer implementation in FPGA with fixed-point resolution analysis on the BER performance and improvement in hardware resource requirement in FPGA implementation. To our knowledge this is the first time that the performance metric of DNN equalizer for optical communication link is reported over varying fixed-point resolutions and corresponding hardware resource usages in actual FPGA implementations. We also investigate joint classification output stages and verify that they can help us recover sensitivity penalty when the fixed resolution is low. While this work is limited up to two embedded parallel output per DNN equalizer, the techniques and results presented here can help us extend the number of parallel outputs beyond two without linearly increasing hardware complexity.

## Acknowledgment

The authors would like to thank Yannick Lefevre, Mathieu Chagnon, Chenhui Ye, Dongxu Zhang and Jochen Maes for valuable discussions.

## References

[1] 25GS PON Specification V2.0, 25GS MSA Group, 2021.
[2] IEEE Standard for Ethernet Amendment 9: Physical Layer Specifications and Management Parameters for 25 Gb/s and 50 Gb/s Passive Optical Networks, IEEE Standard 802.3ca-2020, 2020.
[3] G.9804.3: G.hsp.50Gpmd Draft, ITU-T Standard Group 15 Q2/15, 2021.
[4] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Adaptive equalization of finite non-linear channels using multilayer perceptrons," *Signal Process.*, vol. 20, no. 2, pp. 107–119, Jun. 1990, doi: 10.1016/0165-1684(90)90122-F.
[5] B. Karanov *et al.*, "End-to-end deep learning of optical fiber communications," *J. Lightw. Technol.*, vol. 36, no. 20, pp. 4843–4855, Oct. 2018, doi: 10.1109/JLT.2018.2865109.
[6] M. Chagnon, J. Siirtola, T. Rissa, and A. Verma, "Quantized deep neural network empowering an IM-DD link running in realtime on a field programmable gate array," in *Proc. 45th Eur. Conf. Opt. Commun.*, 2019, p. 4, doi: 10.1049/cp.2019.1190.
[7] V. Houtsma, E. Chou, and D. van Veen, "92 and 50 Gbps TDM-PON using neural network enabled receiver equalization specialized for PON," in *Proc. Opt. InfoBase Conf. Papers*, 2019, pp. 18–20.
[8] N. Kaneda *et al.*, "FPGA implementation of deep neural network based equalizers for high-speed PON," *Opt. InfoBase Conf. Papers*, vol. Part F174, pp. 2020–2022, 2020, doi: 10.1364/OFC.2020.T4D.2.
[9] I. Goodfelllow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: The MIT Press, 2016.
[10] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Hoboken, NJ, USA: Prentice Hall, 1998.
[11] N. Kaneda, D. Van Veen, A. Mahadevan, and V. Houtsma, "DSP for 50G/100G hybrid modulated TDM-PON," in *Proc. Eur. Conf. Opt. Commun.*, 2020, pp. 1–4, doi: 10.1109/ECOC48923.2020.9333248.
[12] N. Kaneda, "Optimizing optical interconnects: Digital predistortion and postequalization techniques in optical communications," *IEEE Microw. Mag.*, vol. 20, no. 5, pp. 98–114, May 2019, doi: 10.1109/MMM.2019.2898023.
[13] I. Stamoulias, K. Georgoulakis, S. Blionas, and G. O. Glentis, "FPGA implementation of an MLSE equalizer in 10Gb/s optical links," in *Proc. Int. Conf. Digit. Signal Process.*, 2015, pp. 794–798, doi: 10.1109/ICDSP.2015.7251985.

[14] T. Suzuki, S. Y. Kim, J. I. Kani, and J. Terada, "Real-Time implementation of coherent receiver DSP adopting stream split assignment on GPU for flexible optical access systems," *J. Lightw. Technol.*, vol. 38, no. 3, pp. 668–675, Feb. 2020, doi: 10.1109/JLT.2019.2950155.

[15] N. Kaneda and A. Leven, "Coherent polarization-division-multiplexed QPSK receiver with fractionally spaced CMA for PMD compensation," *IEEE Photon. Technol. Lett.*, vol. 21, no. 4, pp. 203–205, Feb. 2009, doi: 10.1109/LPT.2008.2009772.

[16] N. Kaneda, Q. Yang, X. Liu, S. Chandrasekhar, W. Shieh, and Y.-K. Chen, "Real-Time 2.5 GS/s coherent optical receiver for 53.3-Gb/s sub-banded OFDM," *J. Lightw. Technol.*, vol. 28, no. 4, pp. 494–501, Apr. 2010, doi: 10.1109/JLT.2009.2031126.

[17] F. Li, X. Xiao, X. Li, and Z. Dong, "Real-time demonstration of DMT-based DDO-OFDM transmission and reception at 50Gb/s," in *Proc. 39th Eur. Conf. Exhib. Opt. Commun.*, 2013, pp. 1–3, doi: 10.1049/cp.2013.1658.

[18] N. Kaneda *et al.*, "Field demonstration of 100-Gb/s real-time coherent optical OFDM detection," *J. Lightw. Technol.*, vol. 33, no. 7, pp. 1365–1372, Apr. 2015, doi: 10.1109/JLT.2015.2399771.

[19] B. Baeuerle, A. Josten, M. Eppenberger, D. Hillerkuss, and J. Leuthold, "Low-complexity real-time receiver for coherent Nyquist-FDM signals," *J. Lightw. Technol.*, vol. 36, no. 24, pp. 5728–5737, Dec. 2018, doi: 10.1109/JLT.2018.2877479.

[20] R. Elschner, F. Frey, C. Schmidt-Langhorst, J. K. Fischer, and C. Schubert, "FPGA implementation of a data-aided single-carrier frequency-domain equalizer for format-flexible receivers," in *Proc. Photon. Netw.; 16. ITG Symp.*, 2015, pp. 1–6.

[21] L. Liu *et al.*, "FPGA- based implementation of artificial neural network for nonlinear signal-to-noise ratio estimation," in *Proc. Optoelectron. Commun. Conf.*, 2021, vol. 1, pp. 4–6.

[22] A. Tomasov, T. Horvath, P. Munster, M. Holik, and V. Oujezsky, "FPGA xPON traffic analysis," in *Proc. 44th Int. Conf. Telecommun. Signal Process.*, 2021, pp. 58–61, doi: 10.1109/TSP52935.2021.9522610.

[23] X. Huang, D. Zhang, X. Hu, C. Ye, and K. Zhang, "Recurrent neural network based equalizer with embedded parallelization for 100Gbps /λ PON," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, 2021, pp. 3–5.

[24] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Apr. 2017, doi: 10.1109/TCCN.2017.2758370.

[25] M. Bahoura, "Efficient FPGA-based architecture of the overlap-add method for short-time fourier analysis/synthesis," *Electron,* vol. 8, no. 12, 2019, Art. no. 1533, doi: 10.3390/electronics8121533.

[26] S. Jalali, C. Nuzman, and I. Saniee, "Efficient deep learning of GMMs," *CoRR*, vol. abs/1902.05707, pp. 1–25, May 2019.

[27] Xilinx Inc., "Zynq ultrascale + MPSoC overview," *Xilinx*, vol. DS891, pp. 1–41, 2021.

[28] Xilinx, *PetaLinux Tools Documentation Reference Guide*, vol. 1144. San Jose, CA, USA: Xilinx, 2019, pp. 1–144.

[29] L. Shu, J. Li, Z. Wan, W. Zhang, S. Fu, and K. Xu, "Overestimation trap of artificial neural network: Learning the rule of PRBS," in *Proc. Eur. Conf. Opt. Commun.*, 2018, pp. 1–3, doi: 10.1109/ECOC.2018.8535327.

[30] E. Harstead *et al.*, "From 25 Gb/s to 50 Gb/s TDM PON: Transceiver architectures, their performance, standardization aspects, and cost modeling," *J. Opt. Commun. Netw.*, vol. 12, no. 9, pp. D17–D26, 2020, doi: 10.1364/JOCN.391945.

[31] Xilinx Inc., "Introduction to FPGA design with vivado high-level synthesis UG998," *Ug998*, vol. UG998, pp. 1–89, 2019.

[32] Xilinx Inc., "Vivado design suite user guide," *Ug903*, vol. 4, pp. 1–173, 2015.

[33] C. Ye, X. Hu, D. Zhang, and K. Zhang, "A versatile NN-equalization for 50Gbps TDM PON burst uplink," in *Proc. Eur. Conf. Opt. Commun.*, 2020, pp. 2020–2022, doi: 10.1109/ECOC48923.2020.9333360.