CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Real-Time FPGA System for Modulation Format Identification of Optical Fiber Communication

Master's thesis in Embedded Electronic System Design

Candi Wu

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2021

Master's thesis 2021

Real-Time FPGA System for Modulation Format Identification of Optical Fiber Communication

Candi Wu

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2021

Real-Time FPGA System for Modulation Format Identification of Optical Fiber Communication
Candi Wu

Supervisor: Per Larsson-Edefors, Department of Computer Science and Engineering
Examiner: Lena Peterson, Department of Computer Science and Engineering

Master's Thesis 2021
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2021

Real-Time FPGA System for Modulation Format Identification of Optical Fiber Communication
Candi Wu
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

In flexible optical fiber communication systems, modulation format identification (MFI) is a useful feature. Although many MFI methods have been proposed, most of them are suitable for fixed channel conditions only. In addition, there exists no hardware-based implementation for identifying more than two formats. In this thesis, two methods of real-time identification are implemented and evaluated. Both of them passed software and hardware test. The method based on calculating information entropy has the advantages of low hardware overhead and fast processing speed, and another method using artificial neural network (ANN) has the advantages of high accuracy and strong impairments tolerance. In the actual optical fiber communication system, different methods can be selected according to the characteristics of this system.

Keywords: fiber optical communication, modulation format identification, neural network, phase noise, linewidth, chromatic dispersion, FPGA

Acknowledgements

I would like to express my deepest appreciation to my supervisors Prof. Per Larsson-Edefors. He extended a great amount of assistance through out the project and was always patient with my problems. I'm deeply indebted to Erik Börjeson, for much appreciated help with the building up the simulation environment and valuable discussion. I'm also grateful to Magnus Karlsson for providing many insightful suggestions in the area of fiber-optic communication systems. Finally, I would like to thank Lena Peterson for helpful comments on both the languange and content of this thesis.

Candi Wu, Gothenburg, October 2021

Contents

List of Acronyms

AH . . . . . amplitude histogram
ANN . . . . . artificial neural network
ASE . . . . . amplifier spontaneous emission
AWGN . . . . . additive white guassian noise

BPSK . . . . . binary phase shift keying

CD . . . . . chromatic dispersion
CHOICE . . . . Chalmers optical fiber channel emulator
CMA . . . . . constant modulus algorithm
CNN . . . . . convolutional neural network

DSP . . . . . digital signal processing

FPGA . . . . . field-programmable gate array

IE . . . . . information entropy

LUT . . . . . lookup table

MFI . . . . . modulation format identification
MPSK . . . . . multiple phase shift keying
MQAM . . . . . multiple quadrature amplitude modulation

NN . . . . . neural network

OSNR . . . . . optical signal-to-noise ratio

PMD . . . . . polarization mode dispersion
PN . . . . . phase noise
PWL . . . . . piecewise linear

QPSK . . . . . quadrature phase shift keying

RRC . . . . . root-raised-cosine filter

16QAM . . . . . 16 quadrature amplitude modulation
32QAM . . . . . 32 quadrature amplitude modulation
64QAM . . . . . 64 quadrature amplitude modulation
8PSK . . . . . 8 phase shift keying

1

Introduction

Optical fiber communication technology uses light waves as the transmission carrier and optical fiber as the transmission medium. This technology has attracted much attention in recent years because of its advantages of anti-interference, good confidentiality and large bandwidth. However, as user requirements continue to increase, new dynamic optical fiber communication networks are proposed [1]. The dynamic network means that the modulation format and bit rate can be changed freely in the communication.

This dynamic change can improve network flexibility and efficiency. When the user's requirements change or the channel changes, the transceivers can change the modulation formats to ensure communication quality. For example, when the user needs faster transmission speed, a higher-order modulation format can be selected, and a lower-order modulation mode can be selected when the channel suffers from different impairments and the noise increases. In addition, the independent choice of modulation format can also enable the appropriate carrier recovery module to be selected among other functionalities [2]. The flexible selection of the modulation formats of the transmitter requires that the receiver can identify the modulation formats of the unknown signal.

Generally speaking, identification is to obtain some characteristic parameters by processing the signal, and then use a certain method to distinguish the characteristic parameters, so as to determine the modulation format and other parameters of the signal. Modulation format identification (MFI) in receiver [3] is a module between signal detection and signal demodulation. The signal used for identification is not ideal but it suffers from different impairments, and the MFI module needs to tolerate these impairments.

In addition, in optical communication equipment, MFI also plays an important role in the optical performance monitoring devices of intermediate network nodes. The information of the modulation format can either be obtained by prior knowledge or from an upper-layer protocol [4]. But the optical performance monitor cannot obtain this information from cross-layer communication because the processing ability at the node is limited [2].

1.1 Aim

Most of the current research focuses on new MFI algorithms, there is only one discussion on the hardware implementation of identifying two modulation formats [5].

The goal of this thesis is to use field-programmable gate array (FPGA) to realize real-time MFI for four commonly used modulation formats at the receiving end. In contrast to the general research direction, the focus here is to find one MFI method that is easy to implement, tolerant to multiple impairments, and keeps the hardware overhead as small as possible. Some high-accuracy methods will be selected for comparison with each other, and the architecture will be optimized and implemented on the hardware to test the real-time processing capabilities.

1.2 Outline

The outline of this report is as follows. First, chapter 2 shows the background and recent research on MFI. Then it introduces 3 MFI methods in detail. These 3 methods will be simulated and discussed for further hardware implementation. Chapter 3 then talks about how the simulation environment is set up. Further, it shows the simulation result for 3 selected methods under different conditions. Chapter 4 contains the architecture of selected methods. Chapter 5 gives the hardware simulation result of selected methods, and also discuss the FPGA resource usage and test accuracy. Chapter 6 summarizes the project and shows future work in the end.

2

Background

The chapter first gives an short introduction to fiber optical communication systems, then introduces existing MFI methods and focuses on three methods implemented in this report. At the end of this chapter, hardware implementation of neural network(NN) will be discussed.

2.1 Optical Fiber Communication System

The optical fiber communication system is shown in Fig. 2.1. The information source contains the digital information to be transmitted. The optical transmitter converts the electrical signal into an optical signal and sends it to channel. The channel contains optical fiber and other optional components such as the optical amplifiers and couplers. The optical receiver includes a coherent receiver and a digital signal processing (DSP) module, after which the information will be transmitted to the information recipient[4].

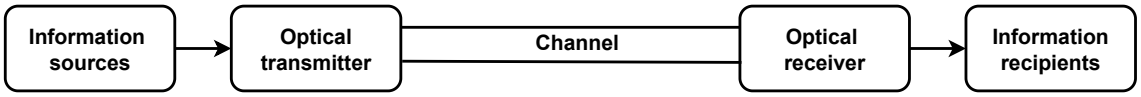

Figure 2.1: An overview of an optical fiber communications link.

2.1.1 Optical Impairments

The influence of optical impairments is reflected in the change of signal amplitude, phase and polarization. Generally, linear impairments are usually considered, typically amplifier spontaneous emission (ASE) noise, chromatic dispersion (CD), polarization mode dispersion (PMD), and phase noise (PN).

ASE noise comes from the optical amplifiers in the transmitter and in the channel. In these amplifieres, stimulated emission occurs when the activated particles interact with incident photons of a specific frequency, emitting new photons with the same phase, frequency, polarization and propagation direction as the incident light [6]. But at the same time, the activated particles can also spontaneously drop to a lower energy level, and emit incoherent light [7]. Because the number of activated particles is constant, the more spontaneous emission photons, the less the activated emission photons, results in less amplified power. Because only this noise affects the amplitude of the signal, the optical signal-to-noise ratio (OSNR) that compares the

power of the desired signal to the power of background noise is used to represent the magnitude of this noise.

CD comes from the phase change caused by the transmission speed of the optical signal [8]. The optical signal emitted by the light source itself may have a bandwidth, and the optical fiber can also cause the dispersion of the optical signal. Fiber dispersion is divided into material dispersion and waveguide dispersion. Material dispersion comes from that refractive index of silica used to make fiber varies with the frequency of optical signal. Therefore, when the optical signal has a bandwidth, the transmission of the optical signal is asynchronous. Waveguide dispersion comes from the geometric structure of the optical fiber, which ultimately leads to different wavelengths of light travel different path lengths. CD only affects the phase of the light and can be measured by the group velocity dispersion parameter D.

PMD is caused by the asymmetry of the optical fiber [8]. When two orthogonal polarization components cannot propagate synchronously, the phase and polarization of the optical signal will change.

PN originates from the laser in the transmitter and the local oscillator in the receiver. This noise can be measured by linewidth [8]. The smaller the linewidth is, the better quality the laser has. PN has no effect on the performance of optical systems using direct receivers, because direct receiving only sensitive to light intensity. Like CD, phase noise only affects the phase of the optical signal.

2.1.2 Optical Receiver

The structure of a common coherent optical receiver is shown in Fig. 2.2. The DSP module first performs analog-to-digital conversion and re-sampling of the acquired analog signal. After that, because I and Q channels of the received signal cannot be absolutely orthogonal, the signal needs to be orthogonalized to restore the original signal. CD is one of channel impairments, and can be compensated by a CD equalizer. Subsequent timing recovery and constant modulus algorithm (CMA) equalization determine the length of each symbol and eliminate inter-symbol interference. The two DSP modules after CMA equalizer will use the modulation format to obtain the phase information between the symbols to calculating frequency offset and remove the phase noise. The last step of DSP is forward error correction, then the signal will be sent to recipient.

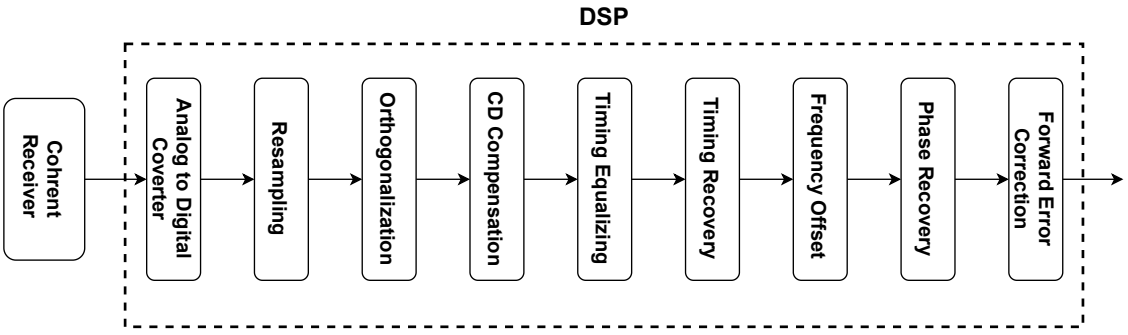

Figure 2.2: Block diagram of a common optical receiver.

2.2 Modulation Format

Information needs to be modulated before being transmitted to become a signal suitable for transmission. Multiple phase shift keying (MPSK) and multiple quadrature amplitude modulation (MQAM) are commonly used modulation formats in digital communications, and they are also the most frequently involved identification objects in MFI [3, 9, 10, 11]. MPSK is a phase modulation method. The amplitude and frequency are taken as constants during the modulation process. The carrier phase will switch between two different values according to the two levels of the digital baseband signal. M represents how to divided 2π equally. Generally, binary phase shift keying (BPSK), quadrature phase shift keying (QPSK), and 8 phase shift keying (8PSK) are commonly used. MQAM is a technique of joint amplitude and phase modulation. Information bits are transmitted through the amplitude and phase of the carrier. The modulated signal can be regarded as the sum of two mutually orthogonal carrier modulation signals. This method has very high spectrum efficiency. M here has the same meaning as the previous. Generally, 16 quadrature amplitude modulation (16QAM), 32 quadrature amplitude modulation (32QAM) and 64 quadrature amplitude modulation (64QAM) are commonly used.

This project will involve four modulation formats: QPSK, 16QAM, 32QAM and 64QAM. The corresponding constellation diagram after the signal is modulated is shown in Fig. 2.3. The points in the complex plane include the phase and amplitude information of the modulated symbols.

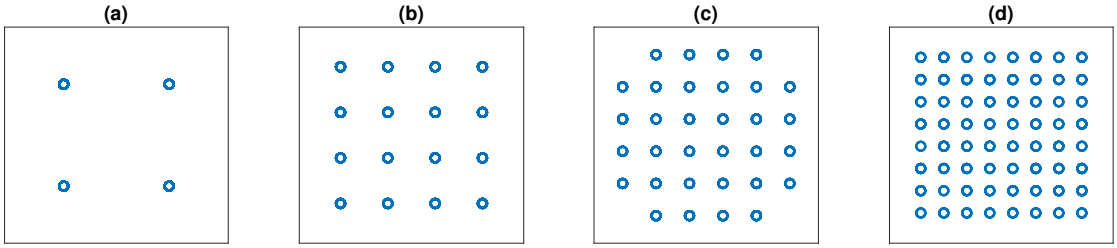

Figure 2.3: Constellation diagram of (a) QPSK, (b) 16QAM, (c) 32QAM, (d) 64QAM.

2.3 MFI Methods

As introduced in previous section, the recovery of phase noise requires the information of modulation formats, so MFI will be added before calculating frequency offset. In this step, most of CD is compensated, but there are still ASE noise and phase noise. The MFI method should be able to tolerate these three noise types. The MFI methods that satisfy the requirements can be roughly divided into two categories: One category is to use feature extraction to realize a blind recognition of the signal. The other one is to use probability theory and the likelihood method [12].

The selection of the characteristics of the first category is very important. The complexity and accuracy vary with different features. In the literature, one research

work [3] is based on analyzing the information entropy of signal's amplitude. The computational complexity of this method is very low and it can identify commonly used modulation modes. In addition, there are methods of identifying with amplitude circle number [10] and Fourier series fitting method [9], which also has a very low computational complexity. In the design of the classifier, some researchers are now committed to combining machine learning and MFI. One study has proposed that the variance of the received signal and OSNR can be used to train a neural network [13]. Other researchers use a deep neural network to distinguish modulation formats based on amplitude histogram [11]. There are also researchers who directly feed a wireless signal into training, allowing allow automatic learning [14]. The second category is based on testing hypotheses and Bayesian methods as the theoretical basis, and the optimal solution can be obtained when the error classification probability is the smallest [15].

Generally, the second category is more complex, but the accuracy is not higher than the first category [12]. Therefore, after comparing the complexity of hardware implementation in the first category of methods, three methods using characteristics extraction were selected for further comparison.

2.3.1 Method 1: Information Entropy Based Method

The first method is to use the information entropy (IE) of the received signal amplitude as a selected characteristic to identify the modulation format [3]. As shown in Fig. 2.4, QPSK, 16QAM, 32QAM, and 64QAM signals correspond to different amplitude distributions.

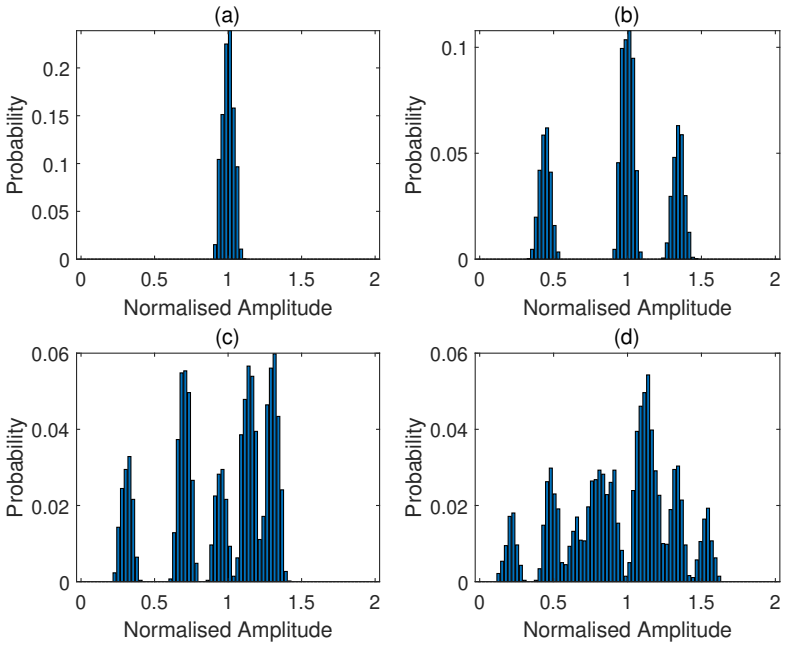

Figure 2.4: Amplitude histogram of (a) QPSK, (b) 16QAM, (c) 32QAM, (d) 64QAM when OSNR is 20 dB.

If the x-axis in Fig. 2.4 is divided into several bins, then for different modulation formats, the probability of the amplitude of the received signal falling on these bins is different. Eq.(2.1) gives a method to calculate information entropy, where p_i is the probability of falling in different areas.

$$H(x) = \sum_{i=1}^{n} -p_i(x) \log_k p_i(x) \tag{2.1}$$

It can be seen that when the number of bins, n, changes, the probability value corresponding to different bins changes, and the calculated information entropy also changes accordingly. As shown in Table 2.1, when the number of bins is 7, the difference in information entropy corresponding to different modulation formats is the largest. So in this method, the number of bins is set to 7 for best identification. These different values in Table 2.1 will be used in later chapters as thresholds to identify different modulation formats.

Table 2.1: Information entropy of different modulation formats with different number of bins.

Modulation format	H_3/bit	H_5/bit	H_7/bit	H_9/bit
QPSK	0	0.71	0	0.11
16QAM	1.51	1.96	1.51	1.93
32QAM	1.56	2.26	2	2.5
64QAM	1.47	2.25	2.34	2.66

In the receiver, when a new sample is obtained, the MFI module will update the probability in different bins. After receiving a certain number of symbols, information entropy will be calculated. Finally, the MFI module will predict the modulation format based on different preset thresholds. This method can be applied to 28 GBaud symbol rate in simulation. If 7% forward error correction is used, this system can identify QPSK, 16QAM, 32QAM, and 64QAM at the lowest OSNR of 12 dB, 18 dB, 22 dB, and 24 dB [3]. Because the amplitude is used for calculation, this method is not sensitive to the phase noise caused by the linewidth, and has a certain tolerance for residual CD.

2.3.2 Method 2: ANN-Based Method

The second method is to use an artificial neural network (ANN) as a classifier, and the amplitude histogram (AH) as the feature to identify modulation formats as presented by Khan in a recent paper [11]. The paper claims that this method can be applied to 28 GBaud QPSK, 14 GBaud 16QAM, and 20 GBaud 64QAM when the linewidth is 100 kHz. And the ranges of different modulation formats are QPSK $10 \sim 23$ dB, 16QAM $17 \sim 26$ dB, 64QAM $25 \sim 37$ dB.

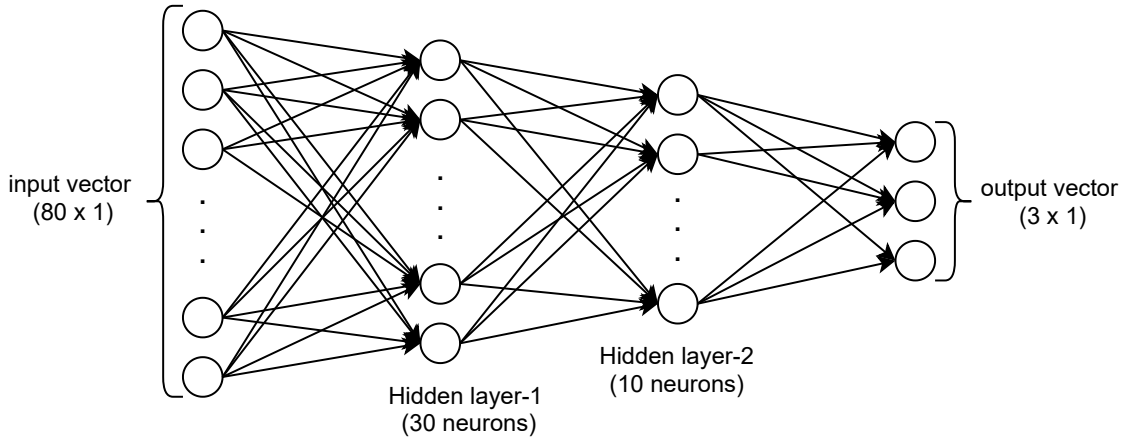

Figure 2.5: Network structure of the proposed two hidden layers ANN [11].

The structure of the neural network proposed is shown in the Fig. 2.5. As the input of the used ANN is the amplitude histogram, the amplitude histogram is transformed to a matrix of size 80×1 where each value is the number of symbols whose amplitude is in the current bin. And the output generated from the ANN is an output matrix of size 3×1, representing the three different modulation formats. The activation function of the two hidden layers is sigmoid [16], and the activation function of the output layer is softmax [16].

2.3.3 Method 3: CNN-Based Method

The hardware complexity of the third method is much higher than that of the first two methods because it is using convolutional neural network (CNN) as a classifier, but it can monitor OSNR while performing MFI. This method uses a constellation diagram as the selected feature and the CNN is as shown in Fig. 2.6 [17]. This method can be used to identify 6 modulation formats with 25 GBaud baudrate, QPSK, 8PSK, 8QAM, 16QAM, 32QAM, and 64QAM. The first five modulation formats can identify OSNR from $15 \sim 30$ dB, and the OSNR range for 64QAM is $20 \sim 35$ dB.

The input data is a grayscale image with a resolution of 28×28. The output data is a matrix of 22×1, where the first six bits represent the different modulation formats, and the last 16 bits represent the estimated value of OSNR. The CNN has 7 layers in total, including one input layer, two convolutional layers, two pooling layers, one full-connection layer, and one output layer.

2.4 Activation Functions

An NN usually contains non-linear activation functions sigmoid and softmax [16]. In FPGA implementation, how to optimize the architecture so that the nonlinear functions can be accurately implemented is a direction of discussion.

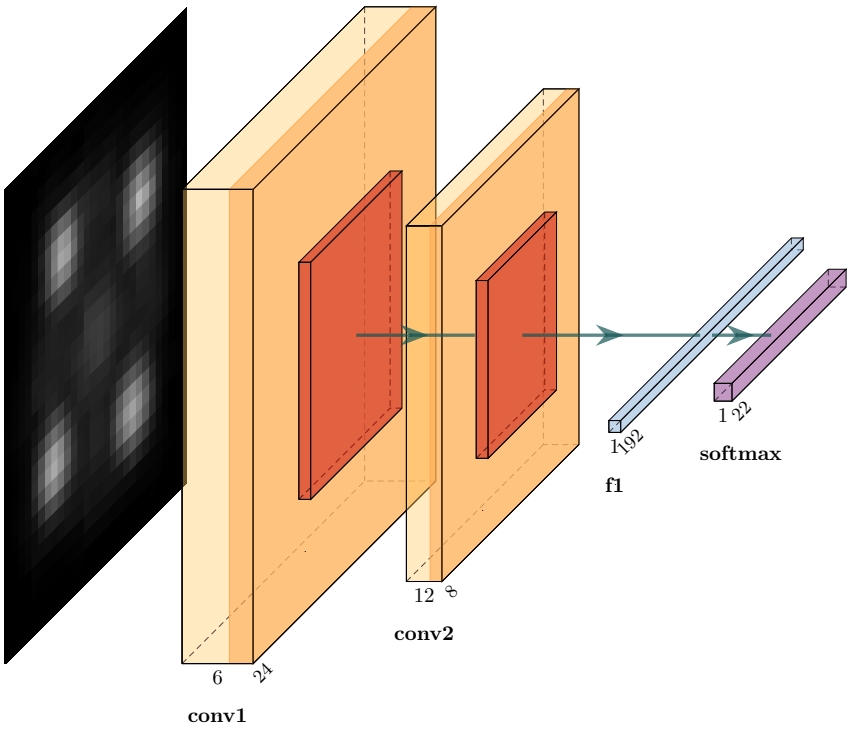

Figure 2.6: Network structure of the proposed CNN [17]. Input layer: 28×28, convolution layer 1 (C1): six 24×24, pooling layer 1 (P1): six 12×12, C2: twelve 8×8, P2: twelve 4×4, Fully connected layer 1 (F1): 192×1, Output layer: 22×1

2.4.1 Sigmoid

The sigmoid function is a common activation function, which can be expressed as $f(x) = \frac{1}{1+e^{-x}}$. The hardware implementation methods of this nonlinear function, in the research literature, are, Taylor series, piecewise linear (PWL) approximation, polynomial fitting, lookup table (LUT), and CORDIC [18, 19, 20, 21]. The way to use the Taylor expansion is to expand the e^{-x}, and the result is $e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + ... + \frac{(-1)^n x^n}{n!}$. By selecting the number of terms used in sigmoid, the accuracy of this function can be determined.

When using PWL approximation to implement the sigmoid function, the choice of function determines the accuracy of the fit. One set of functions having average deviation of 0.01412 are shown in Table 2.2.

Table 2.2: PWL approximation of sigmoid [19].

x	y
$0 \leq \|x\| < 1.0$	$y = 0.25 \cdot \|x\| + 0.5$
$1.0 \leq \|x\| < 2.375$	$y = 0.125 \cdot \|x\| + 0.625$
$2.375 \leq \|x\| < 5$	$y = 0.03125 \cdot \|x\| + 0.84375$
$\|x\| \geq 5$	$y = 1$

Similar to PWL, polynomial fitting also needs a set of equation. When average deviation is 0.00769, the functions can be chosen as Table 2.3. Because sigmoid has the property $f(x) = 1 - f(-x), x < 0$, only the polynomial fitting when x is greater than 0 is listed in the table.

Table 2.3: Polynomial fitting of sigmoid [19].

x	y
$0 \leq x < 4.0$	$y = -0.03125 \cdot x^2 + 0.25 \cdot x + 0.5$
$x \geq 4$	$y = 1$

Another novel structure was proposed in [22]. The input data can be approximated as $n \ln n$, so that the calculation of the exponential function can be transformed into a shift operation. Furthermore, considering the difference between x and $n \ln n$, the sigmoid function can be calculated using Eq. (2.2).

$$\begin{aligned} f(x) &= f(n \ln 2) + f'(n \ln 2) \Delta x \\ &= f(n \ln 2) + (f((n+1) \ln 2) - f(n \ln 2)) \left(\frac{x}{\ln 2} - n \right) \end{aligned} \tag{2.2}$$

In Eq. (2.2), $f(n \ln 2)$ can be obtained by the bit operation of the integer part of the input, $f((n+1) \ln 2) - f(n \ln 2)$ is a series of fixed values, which can be represented by $2^{m1} + 2^{m2}$ and convert the multiplication into shift. The average error of this method can be reduced to 0.0016.

Using LUT is a simple way to implement sigmoid. According to the targeted accuracy, output value of sigmoid is stored in LUT. High precision requires large memory. CORDIC is using hyperbolic function to calculate sigmoid. This method gains accuracy at the cost of processing time. To achieve a deviation lower than 0.005, 50 clock cycles are needed [21].

2.4.2 Softmax

The softmax function is widely used in the output layer of classification problems. The output layers of the two neural networks mentioned in the previous section both use softmax as the activation function. The definition of softmax is $f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} (i = 1, 2, \ldots, N)$. The calculation first needs the exponential value of each input, N is the total number of inputs. Similar to sigmoid, the calculation of exponential function can also be calculated by Taylor series [23], PWL approximation [24], polynomial fitting and LUT [25]. After the exponent value is obtained, the values can be summed and calculated the final probabilities. Unlike sigmoid, when using PWL approximation, polynomial fitting and LUT, the range of the fitted exponential function needs to be determined according to the range of the input data. Because softmax increases dramatically when x is greater than 0, the maximum value of the output can be taken to infinity. One possible choice of PWL function when x is ranging from -2 to 2 is as shown in Table 2.4.

Table 2.4: PWL approximation of softmax [24].

x	y
$-2 \leq x < 0$	$y = 0.4323 \cdot x + 1$
$0 \leq x < 2$	$y = 3.1945 \cdot x + 1$

In addition to the above methods which are similar to sigmoid, researchers have proposed a new method [26].

$$\begin{aligned} f(x_i) &= \exp\left(\ln\left(\frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}\right)\right) \\ &= \exp\left(x_i - \ln\left(\sum_{j=1}^{N} e^{x_j}\right)\right)(i = 1, 2, \ldots, N)\,. \end{aligned} \tag{2.3}$$

Softmax can be transformed into Eq. (2.3). The exponential function can be expressed as $e^{y_i} = 2^{y_i \cdot \log_2 e} = 2^{u_i + v_i} = \begin{cases} 2^{v_i} \ll u_i & u_i > 0 \\ 2^{v_i} \gg (-u_i) & u_i \leq 0 \end{cases}$, where $u_i = y_i \cdot \lfloor \log_2 e \rfloor$ is an integer, and v_i is the decimal part of $y_i \cdot \log_2 e$. When v_i is less than 1, 2^{v_i} can be approximated as $v_i + d$. The logarithmic part of Eq. (2.3) can be expressed as Eq. (2.4), where w and $\log_2 k$ come from $\forall F > 0, \exists!(k, w) : F = 2^w \cdot k, k \in [1, 2)$. w is the position of the highest 1 in the binary number. From the range of k, the logarithmic term can be expressed as $\log_2 k \approx k - 1$. After formulating softmax into the above equations, multipliers and dividers can be replaced by adders and shift registers.

$$\ln F = \ln 2 \cdot \log_2 F = \ln 2(w + \log_2 k)). \tag{2.4}$$

2.5 Chalmers Optical Fiber Channel Emulator

The hardware realization of the optical fiber system involves fixed-point mapping of floating-point operations and a large number of DSP operations. At the same time, an FPGA has limited resources. As the complexity of the system increases, the hardware overhead will gradually increase. The Chalmers optical fiber channel emulator(CHOICE) is an emulator developed based on hardware[27] which gives a good solution for hardware implementation of optical fiber system. This emulator contains most of the components of the optical fiber system used in this project, and can be used in real-time FPGA test. A more detailed introduction to the CHOICE system is in section 4.1.

3

Simulation of MFI Methods

In chapter 2, three different MFI methods are introduced. In this chapter, these three methods will be compared with each other using MATLAB simulations based on the three impairments mentioned in the previous section. These three impairments are affected by three parameters: SNR, residual CD, and linewidth. After discussing the effects of each parameter separately, the worst case for one MFI method can be obtained.

3.1 Residual CD

The implementation of method 1 depends on the chosen thresholds. With the number of symbols is 2500, the linewidth is 100 kHz, the system baudrate is set to 28 GBaud, and parameter D of fiber is 16 ps/(nm×km), as the residual CD changes, the information entropy changes with SNR as shown in Fig. 3.1.

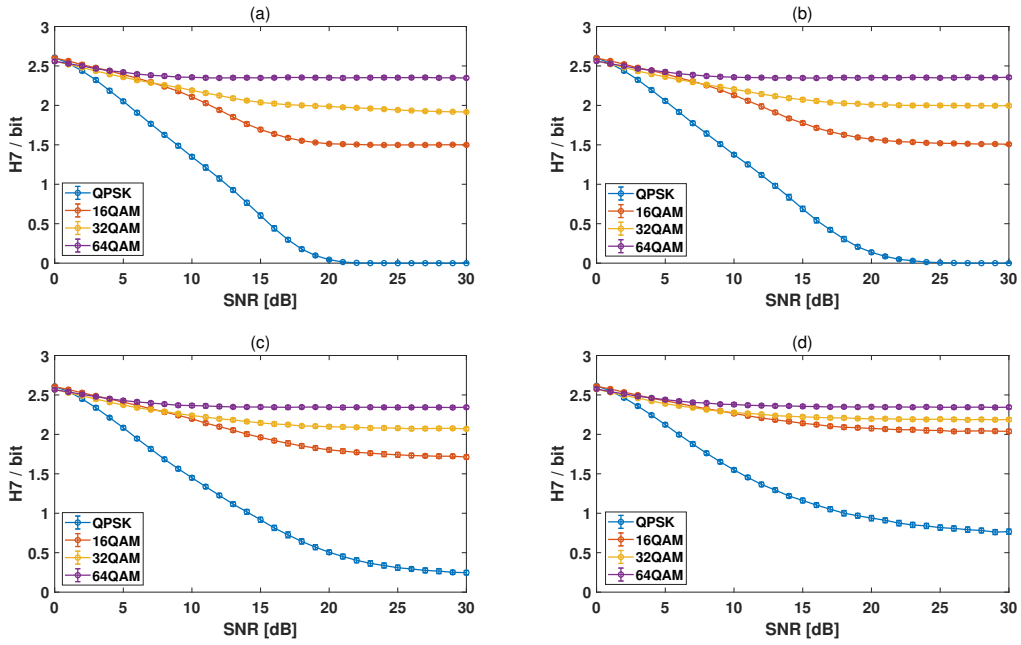

Figure 3.1: The values of H_7 vary with different OSNR when the residual CD is (a) 0 ps/nm, (b) 32 ps/nm, (c) 64 ps/nm, (d) 96 ps/nm in method 1.

As the residual CD continues to increase, the difference between the entropy values of modulation formats decrease. This effect is more obvious in QAM than in QPSK. Considering that there will be more impairments in the real environment, the difference among the information entropy of the four formats will be further reduced. So in this case, this method has a residual CD of 64 ps/nm as the worst case. After this value is exceeded, the curves of information entropy may overlap with each other in a real environment and this method cannot work properly.

Table 3.1: The setting of different impairments in method 2. The SNR and the residual CD are randomly chosen in the given ranges

Format	Baudrate	SNR	Residual CD	Linewidth	Samples
QPSK	28 GBaud	7 ∼ 30 dB	0 ∼ 96 ps/nm	100 kHz	500
16QAM	28 GBaud	11 ∼ 30 dB	0 ∼ 96 ps/nm	100 kHz	500
32QAM	28 GBaud	13 ∼ 30 dB	0 ∼ 96 ps/nm	100 kHz	500
64QAM	28 GBaud	16 ∼ 30 dB	0 ∼ 96 ps/nm	100 kHz	500

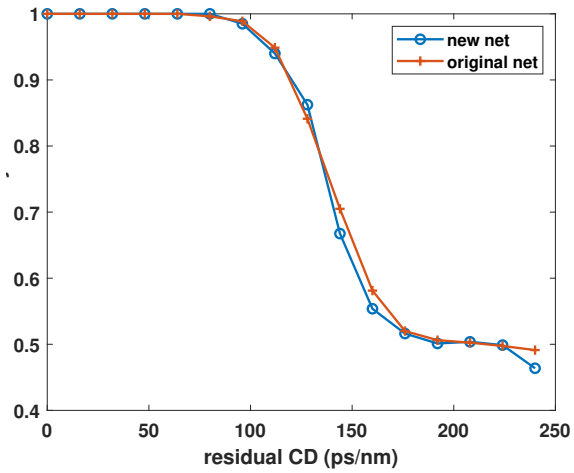

Figure 3.2: The accuracy varies with different residual CD when applying original ANN and new ANN in method 2.

Method 2 uses ANN as a classifier. Before it can be used, a training set needs to be generated for the ANN to be trained. The parameter selected when generating the training set is shown in Table 3.1. Each modulation format has 500 training samples and each sample contains 2500 symbols. The SNR and residual CD in each sample is randomly selected within the range. The range of the residual CD is obtained by testing several different ranges with the same other parameter. Large residual CD can lower the accuracy of identification, so for efficiency and high accuracy, we choose the residual CD in 0 ∼ 128 ps/nm. Considering that when D is 16 ps/(nm×km), the inference set uses the same parameter settings but the residual CD is fixed between 0 ∼ 240 ps/nm and the number of samples in each format is reduced to 200. The accuracy of method 2 is shown by the yellow curve in Fig 3.2. For the residual CD values lower than 112 ps/nm, accuracy is more than 90%.

The network designed in method 2 in the paper [11] has two hidden layers, each with 30 perceptrons and 10 perceptrons. In order to minimize the hardware overhead, we now attempt to reduce the size of the ANN. The size of the ANN can be reduced in two ways: reducing the number of layers, and reducing the number of perceptrons in each layer. In this project, reducing the number of layers greatly reduces the accuracy, so we keep the two hidden layers but instead reducing the number of perceptrons. After trying to train multiple ANNs of different sizes with the same training set, the final size of the ANN was reduced to 10 perceptrons and 5 perceptrons in each layer. The result of this less complex ANN is shown in Fig. 3.2 as the new net (blue line). As shown here, the sensitivity on residual CD does not change significantly as the network is simplified.

Table 3.2: The setting of training sets when residual CD is fixed in method 3. Only SNR is randomly chosen within the given range.

Format	Baudrate	SNR	Linewidth	Samples
QPSK	28 GBaud	$7 \sim 16$ dB	100 kHz	100 per SNR
16QAM	28 GBaud	$11 \sim 20$ dB	100 kHz	100 per SNR
32QAM	28 GBaud	$13 \sim 22$ dB	100 kHz	100 per SNR
64QAM	28 GBaud	$16 \sim 25$ dB	100 kHz	100 per SNR

Table 3.3: The MFI accuracy varies with different residual CD in method 3.

Residual CD	QPSK	16QAM	32QAM	64QAM
0 ps/nm	1	0.99	1	0.97
32 ps/nm	0.99	0.99	1	1
64 ps/nm	1	0.99	1	0.99
96 ps/nm	1	1	1	0.97

Method 3 uses CNN to classify the grayscale image of the constellation diagram. The training set is set as Table 3.2 when the impact of residual CD is studied. The inference sample set also has the same settings as for training set. The monitored OSNR ranges are changed to QPSK $7 \sim 16$ dB, 16QAM $11 \sim 20$ dB, 32 QAM $13 \sim 22$ dB, and 64QAM $16 \sim 25$ dB. The accuracy of method 3 corresponding to different residual CD is shown in Table 3.3. The error of OSNR predicted is shown in Fig. 3.3. It can be seen from the figure that as the residual CD increases, the accuracy of the prediction is constantly decreasing. And the accuracy of predicting high SNR is generally lower than that of predicting low SNR.

3.2 Phase Noise

As mentioned in chapter 2, because method 1 and method 2 use amplitude histograms as input, they are insensitive to PN. This property was also tested during the simulation. When the residual CD is 64 ps/nm and the baudrate is 28 GBaud,

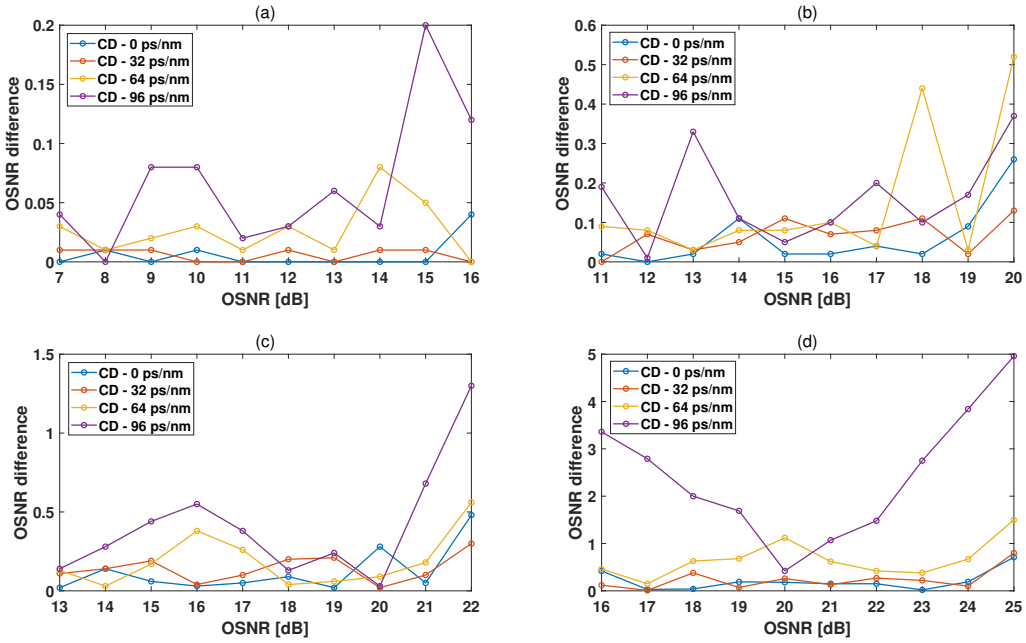

Figure 3.3: The difference between the predicted OSNR and the real OSNR varies with different residual CD in method 3. (a) is QPSK, (b) is 16QAM, (c) is 32QAM, (d) is 64QAM.

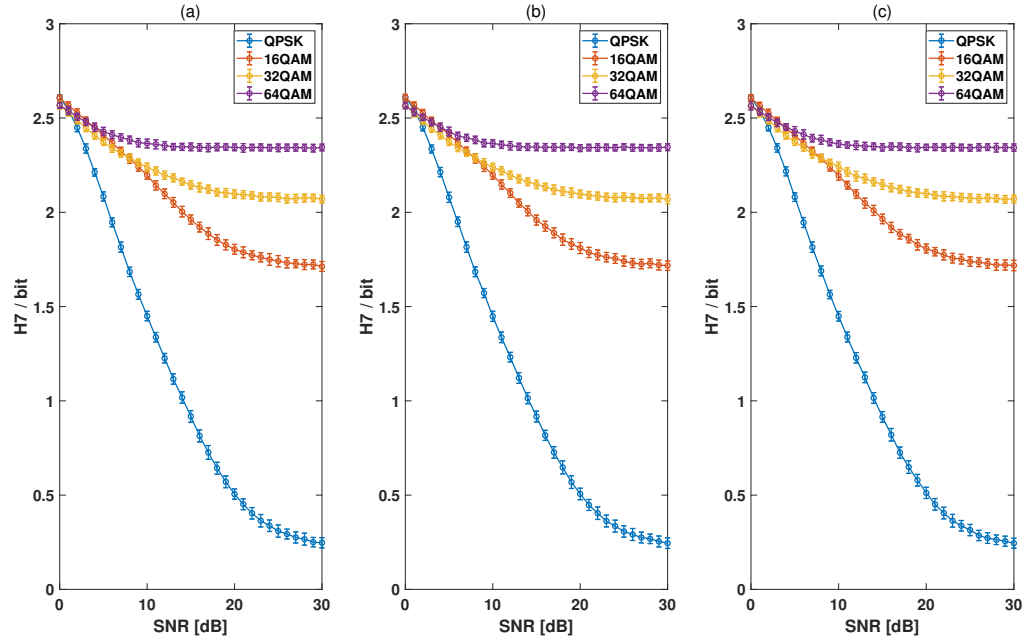

Figure 3.4: The values of H_7 vary with different OSNR when the linewidth is (a) 10 kHz, (b) 100 kHz, (c) 1 MHz in method 1.

the linewidth is changed between 10 kHz, 100 kHz, and 1 MHz in turn. The information entropy of method 1 varies with SNR as shown in Fig. 3.4.

Similarly, method 2 uses the same impairments setting as before but changes the linewidth between 10 kHz, 100 kHz, and 1 MHz. The accuracy changing with residual CD is shown in Fig. 3.5.

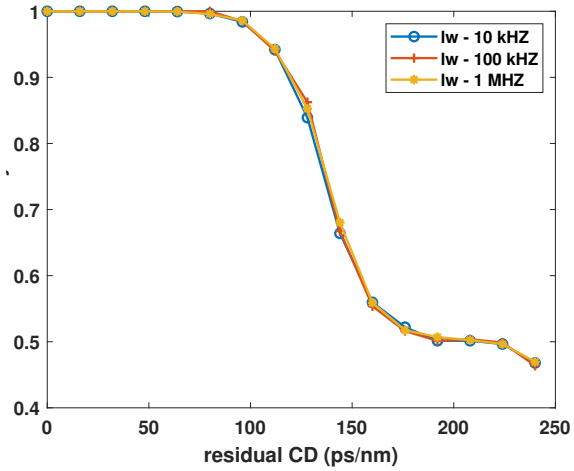

Figure 3.5: The accuracy vary with different residual CD when the linewidth is (a) 10 kHz, (b) 100 kHz, (c) 1 MHz in method 2.

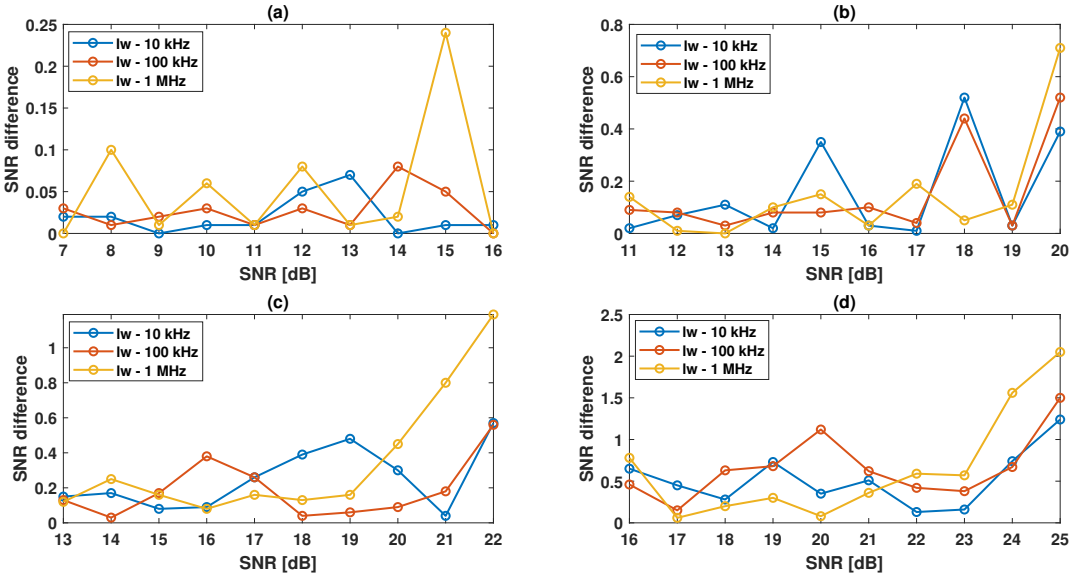

Figure 3.6: The difference between the predicted OSNR and the real OSNR varies with different linewidth in method 3. (a) is QPSK, (b) is 16QAM, (c) is 32QAM, (d) is 64QAM.

The constellation diagram used in method 3 will change with different linewidths. When considering the residual CD of 64 ps/nm and the baudrate of 28 GBaud, the accuracy of the MFI is shown in Table 3.4, and the accuracy of the predicted OSNR is shown in Fig. 3.6. As the linewidth increases, the overall prediction error will increase. From the simulation results obtained above, we find NN-based methods

Table 3.4: The MFI accuracy vary with different residual CD in method 3.

Linewidth	QPSK	16QAM	32QAM	64QAM
10 kHz	1	1	0.99	0.98
100 kHz	1	0.99	1	0.99
1 MHz	1	1	1	0.99

have a better tolerance for residual CD, and the linewidth has little effect on all methods. For the smallest SNR used, because method 1 needs to rely on the selection of the threshold to perform MFI, the lowest SNR value that can be used is not as low as the other two NN-based methods.

4

Architecture

This chapter introduces the specific architectures that were developed for two different MFI methods. While all three methods reviewed show good performance, we select the first two methods for implementation. This is because small area and low power consumption are the main targets of our MFI hardware design. First, this chapter introduces the hardware structure of the entire optical fiber system, and then the specific implementation of the two MFI methods are shown. All components have detailed descriptions.

4.1 Overview

A complete system is shown in Fig. 4.1, which has all corresponding parts to the previous simulation on MATLAB. The transmitter includes a random-number generator (RNG), a modulator and a root-raised-cosine filter (RRC). The channel impairments include additive white gaussian noise (AWGN), phase noise (PN) and CD. AWGN module generates ASE noise as it has the same distribution. Both two modules are from the CHOICE system except for the CD which is developed in this project and which will be described in detail in section 4.1.1. As shown in Fig. 4.1, each impairment is an independent module, which can be added freely. And also, the parameters that control impairments are settable. The MFI block can work under a lower frequency in the system to save power in a practical situation.

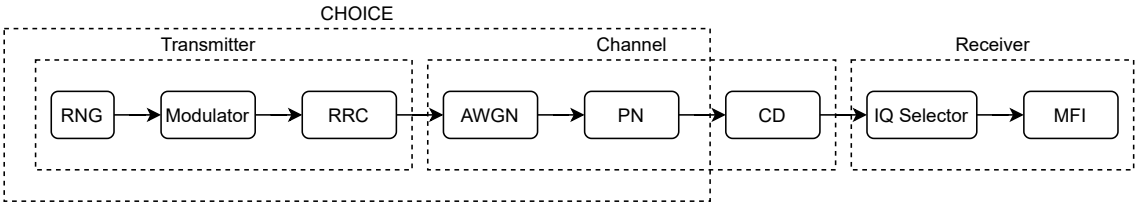

Figure 4.1: The overview of an optical fiber system in hardware environment.

The period of the random bits in RNG is $2^{64} - 1$, and a 12-bit sequence is generated in each clock cycle. Then the sequence is modulated. The modulation formats which are allowable are QPSK, 16QAM, 32QAM and 64QAM. The modulator in MATLAB follows the normalization of average power when generating symbols. But in the VHDL design, in order to reduce the hardware overhead, the design only limits the maximum amplitude of the symbols. When the maximum amplitude is

set to 1, the I and Q components can be represented by a signed 8-bit data. After modulator, the signal goes to an RRC to simulate the process of reducing inter-symbol interference in the real transmission. At the same time, it is upsampled to generate two samples per symbol. The parameters used in AWGN, CD, and PN are calculated in MATLAB and placed in the LUTs for quick access. Thus, the parameters can be changed during VHDL simulation or FPGA emulation. The signals transmitted from the channel to the MFI are one-channel data, and provides two data in the I and Q directions respectively within one clock cycle.

4.1.1 CD

The transfer function of CD in the time domain can be expressed as Eq. (4.1), where D represents the dispersion coefficient of the fiber, λ is the wavelength, and z is transmitted distance [28].

$$h_{CD} = \sqrt{\frac{c}{jD\lambda^2 z}} \exp\left(\frac{jc\pi}{D\lambda^2 z} t^2\right) \tag{4.1}$$

According to chapter 2, the CD impairment comes from the residual electromagnetic waves of the previous symbols. The longer the distance, the more influence from the previous symbols is received. In order to emulate a practical system, to find the residual CD we would need to calculate the long-distance CD on the channel and perform CD compensation at the receiver. But for simplicity, we directly use the short-distance CD instead.

$$(I + jQ) \cdot (h_I + jh_Q) = (Ih_I - Qh_Q) + j(Ih_Q + Ih_Q) \tag{4.2}$$

The transfer function can be replaced by FIR filters. The number of taps is determined by the residual CD. A large residual CD requires large taps to ensure that the output is similar from the convolution of the time domain system function. Because the I and Q components of the signal can affect each other as Eq. (4.2), Four FIR filters are needed. The structure is shown in Fig. 4.2.

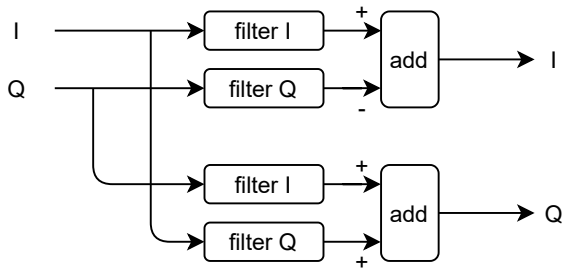

Figure 4.2: The structure for CD module.

4.2 Architecture for Method 1

The implementation of method 1 is composed of two modules as shown in Fig. 4.3. First, the number of symbols whose amplitudes fall in given bins are counted. This

process produces an amplitude histogram (AH) with a low resolution. And then, the possibilities of each bin and information entropy are calculated. The architecture of this method is relatively simple, and the entire process is sequential.

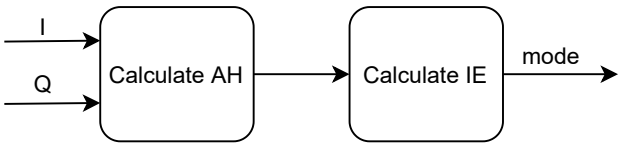

Figure 4.3: Block diagram of method 1.

4.2.1 Calculate AH

To generate an AH, we first need to know the amplitude of the symbols. The calculation of amplitude is given in Eq. (4.3). From the formula, the implementation requires two multipliers and a square root module, which is relatively difficult to implement.

$$|I + jQ| = \sqrt{I \cdot I + Q \cdot Q} \tag{4.3}$$

In order to simplify the structure, the alpha max plus beta min algorithm is used here to calculate the amplitude [29]. The algorithm is shown in Eq. (4.4), where max is larger absolute value of the I and Q components, and min is the smaller absolute value. $\alpha_0, \alpha_1, \beta_0$, and β_1 have multiple values to choose from. As mentioned in section 4.1, the CHOICE system has a different way of generating modulated symbols, so the number of bins must be retested during hardware simulation. The number of bins can only be 3, 5, 7, and 9, so accuracy requirements in this process are low. The selected values are shown in Table 4.1, and the corresponding largest error is -2.65%.

$$\begin{aligned} |z| &= \max(|z_0|, |z_1|), \\ |z_0| &= \alpha_0 Max + \beta_0 Min, \\ |z_1| &= \alpha_1 Max + \beta_1 Min. \end{aligned} \tag{4.4}$$

The AH module is composed of a register array. According to the magnitude of the received amplitude, the corresponding register is updated by one. After collecting enough symbols, the value in the register array will be transmitted to next module.

Table 4.1: Parameter in alpha max plus beta min algorithm.

α_0	β_0	α_1	β_1
1	0	7/8	17/32

4.2.2 Calculate IE

As shown in Eq. (2.1), the calculation of information entropy is an accumulation process, and the object is the probability of each bin multiplied by the negative

logarithm value of this probability. In order to reduce the additional resource consumption caused by the probability calculation, the total number of symbols used can be set to 2^n. Therefore, the division operation in binary can be replaced by simple shifts.

The log function can be approximated using the method of calculating $\ln F$ in Eq. (2.4). The structure of the entire information entropy (IE) module is shown in Fig. 4.4. After detecting the highest 1 bit in the detector, w and $k-1$ can be calculated and added together to get the logarithmic result. When finishing calculating the data in all bins, the information entropy can be obtained at the next clock edge.

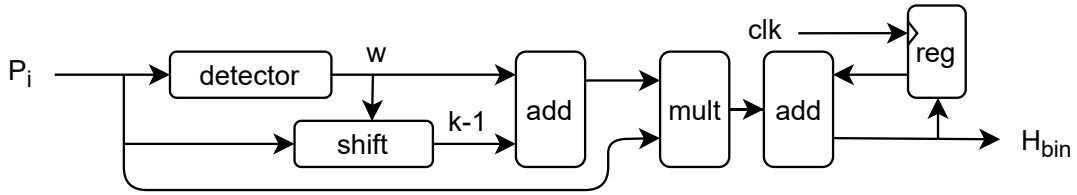

Figure 4.4: Block diagram of module to calculate information entropy.

4.3 Architecture for Method 2

Same as method 1, the input in method 2 is also an amplitude histogram. When the maximum amplitude is limited to 1, the AH of QPSK is more concentrated on the right side of the axis, and the AH of QAMs are more concentrated on the left. While the AH of QPSK has only one peak, there are more than three peaks are in QAMs. So a smaller number of bins can be chosen when drawing the histogram. Smaller AH can retain the characteristics of all modes but reduce the input size and hardware overhead.

Method 2 calculates the AH in the same way as method 1. But because method 2 uses more bins to represent AH, so the accuracy required when calculating the amplitude is higher. The parameters of the alpha max plus beta min algorithm here are set to the values shown in Table 4.2. Compared with method 1, this component has two more multipliers, but the largest error is reduced by 1.52% [29].

Table 4.2: Parameter in alpha max plus beta min algorithm.

α_0	β_0	α_1	β_1
127/128	3/16	27/32	71/128

4.3.1 ANN

The optimized ANN used in this project is shown in Fig. 4.5, including a preprocessing module and a three-layer network. The preprocessing module scales the data between -1 and 1 and the scaling parameters can be obtained from the network

trained by MATLAB. As discussed above, the smaller number of bins can be applied to ANN, so the number of input layers would be adjusted in this hardware design.

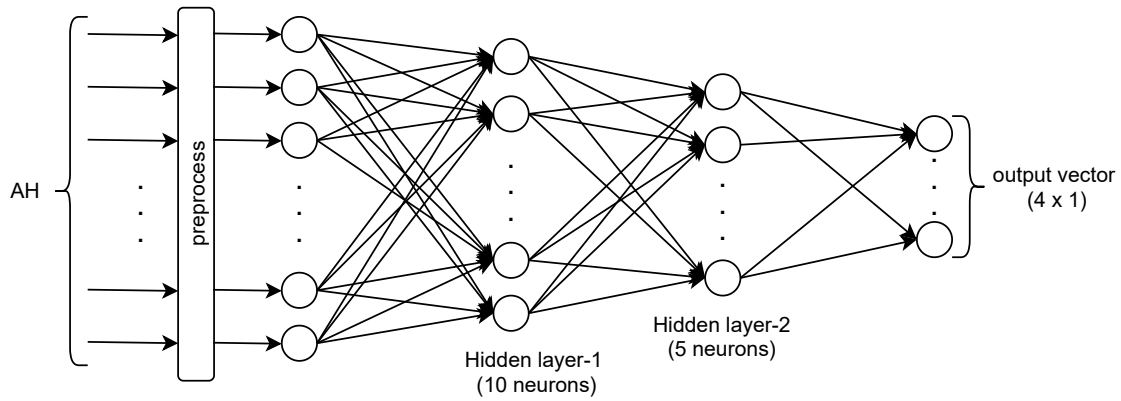

Figure 4.5: Optimized ANN used in method 2.

For each neuron, the internal structure is shown in Fig. 4.6. Each neuron needs to add the weighted inputs to its own bias, and use this value as input to the activation function to obtain the final output. According to this structure, in order to reduce the hardware overhead, we can compute three layers serially and reuse the structure for one layer. The detailed inner structure of the ANN implementation we developed is shown in Fig. 4.7.

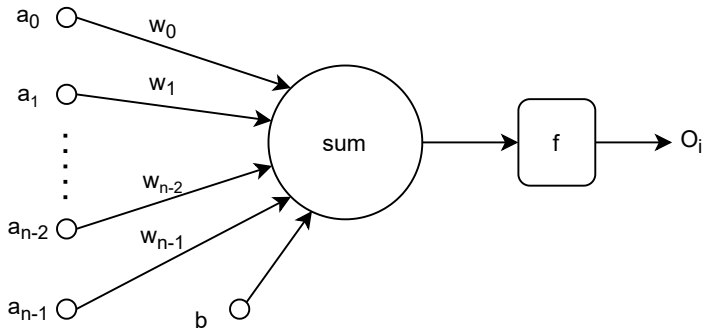

Figure 4.6: Structure of each neuron.

One layer of this neural network can be replaced by 10 12-bit registers. The initial value in the register is the bias of this neuron. Each input from the AH can be sequentially transferred to NN. After preprocessing, the AH input will pass through 10 multiplier arrays, and get added to the data in the registers. After input data looping through all AH bins, the value of the first layer input to the activation function can be obtained. The output value for the activation function will also be stored in 10 12-bit registers. Only one activation function is instantiated, and this instance of activation function is reused for all neurons. Once the first layer is finished, the register storing the output of the activation function will be used as the input to the next layer. The number of neurons in the second hidden layer is 5 neurons fewer than in the first hidden layer. So the unused registers will be set to 0 to reduce the flipping caused by multipliers. The third layer is also calculated

in a similar way, but it uses a different activation function. Unlike the sigmoid, the softmax calculates all the input at the same time.

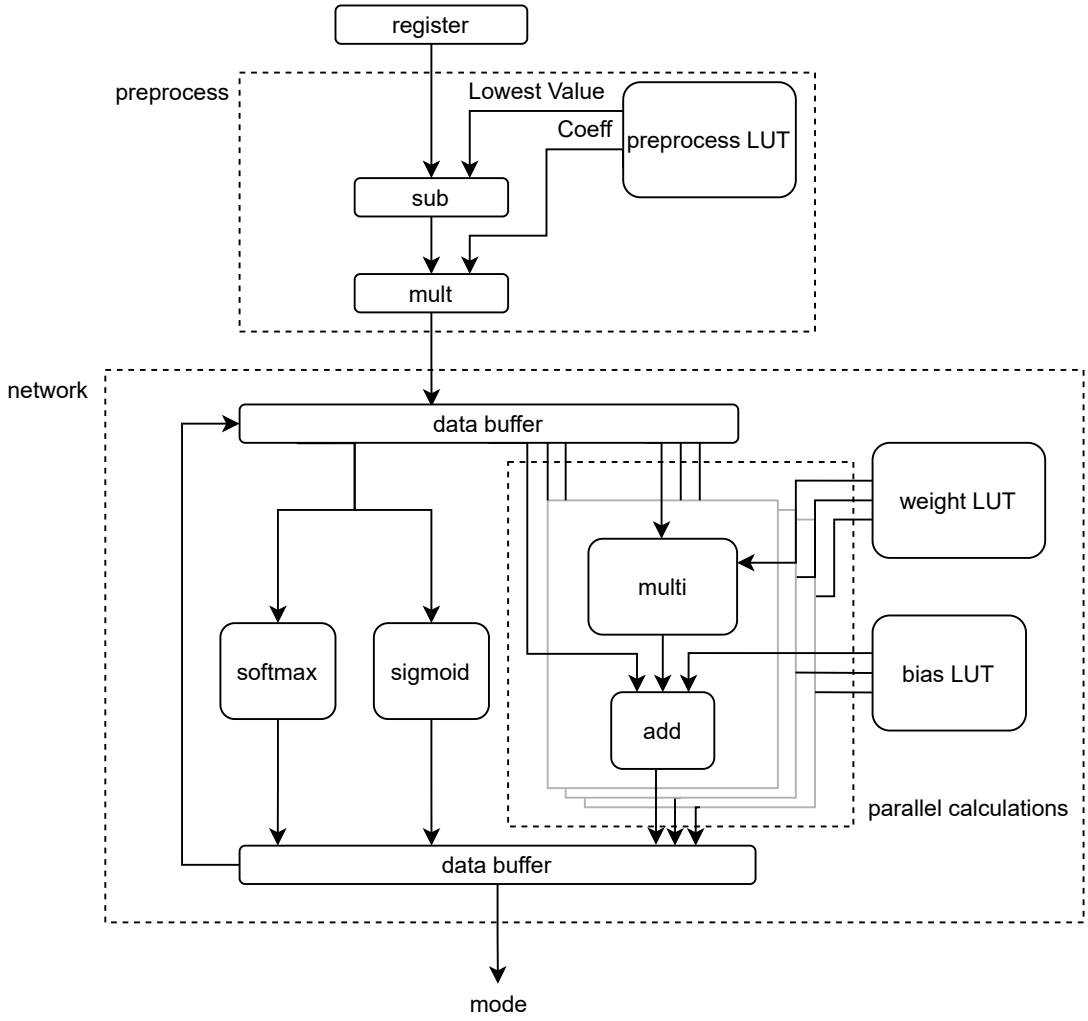

Figure 4.7: Structure of our ANN implementation. All neurons are computed serially to save on hardware resources.

4.3.1.1 The Sigmoid Module

In chapter 2, several methods to realize sigmoid are proposed, and one of them is PWL fitting. In our NN implementation, PWL fitting of Table 2.2 is used to reduce hardware overhead. In Table 2.2, x is multiplied by 2^{-n}, which can be implemented as a right shifter in the hardware, reducing the use of multipliers. For negative input, the properties of $f(x) = 1 - f(x)$ are used. The structure of the sigmoid is shown in Fig. 4.8. The overall structure is relatively simple, and the calculation can be completed within one clock cycle.

4.3.1.2 The Softmax Module

chapter 2 also gives different softmax approximation methods. PWL, which is used in sigmoid implementation, is not suitable for the softmax in our optimised ANN.

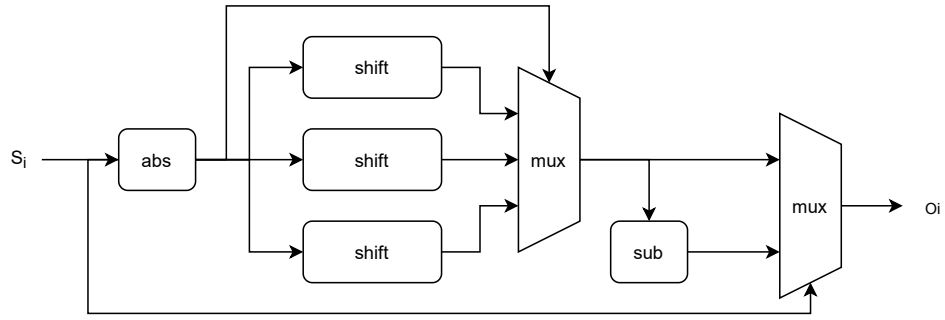

Figure 4.8: Block diagram of sigmoid module.

Because when input data change from 7.9 to −8, PWL requires parameters ranging from 2^{-11} to 2^{11}. Therefore, another method introduced in section 2.4.2 is used here.

This method involves the approximation of exponents and logarithms. The logarithmic calculation is almost the same as calculating IE in method 1, except that it is multiplied by a fixed parameter. The exponential function can also be simplified into a shift after being converted to 2^n. The whole softmax module is shown in Fig. 4.9. Arrows of different colors represent the transmission direction of data in different clocks, and a total of 4 clock cycles are required.

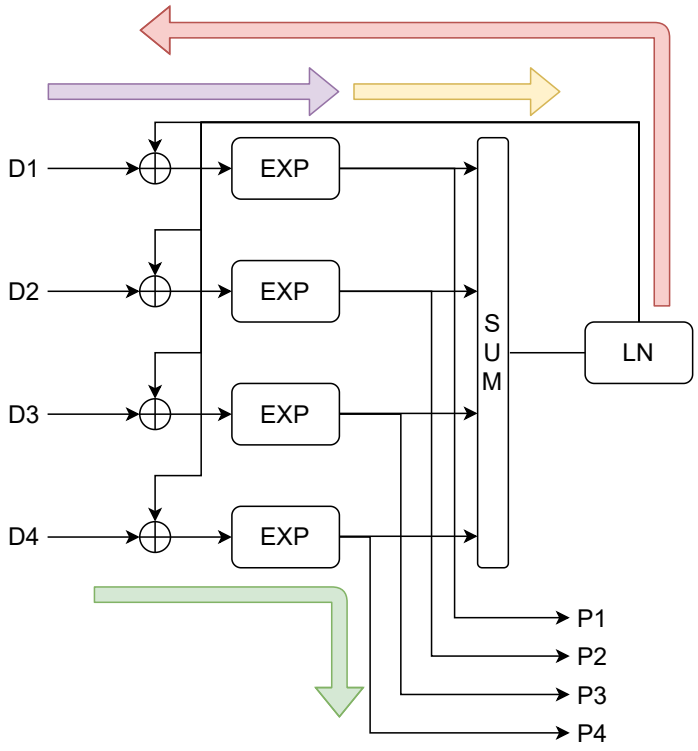

Figure 4.9: Block diagram of the softmax module. Arrows represent the data flow. Inputs first go through the EXP blocks as shown by the purpure arrow, and the outputs are summed up and sent to the LN block as shown by the yellow arrow. After the LN block has done the calculation, data is sent to EXP blocks as shown by red arrow. Finally, the outputs are generated as shown by the green arrow.

5

Result

This chapter presents the logic simulation results produced by the design introduced in Chapter 4, and the test on FPGA. The effect of each impairments will be discussed separately. In addition, the hardware overhead for the complete design will also be evaluated.

5.1 Logic Simulation of Method 1

As mentioned in Section 4.2, when the modulator generates a signal with a maximum amplitude instead of the normalized power, the amplitude histograms of modulation formats change and the corresponding information entropy is also different. Fig. 5.1 shows the information entropy obtained by the simulation when the numbers of bins are 3, 5, 7, and 9.

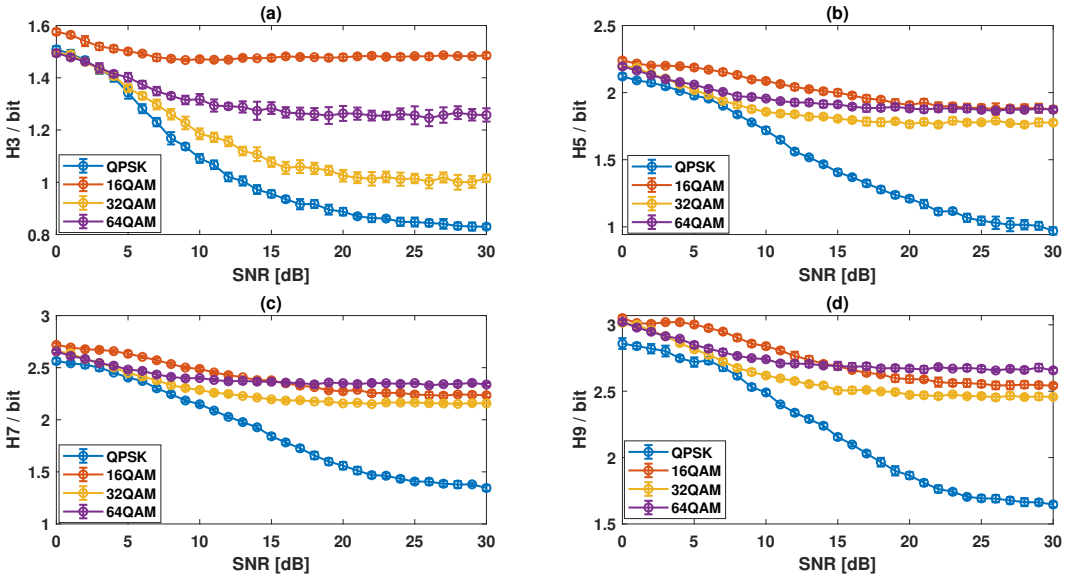

Figure 5.1: The values of information entropy vary with different numbers of bins in method 1. (a) 3 bins, (b) 5 bins, (c) 7 bins, (d) 9 bins.

The information entropy values calculated with 5, 7, and 9 bins overlap and only have small differences when SNR is high. This is because, in hardware implementation, the distribution of peaks of QAMs is more concentrated, high noise can affect the

data corresponding to each bin more. Thus the 7 bins that have been tested well in the MATLAB simulation are not applicable here, and the number of bins will be set to 3 in the following tests.

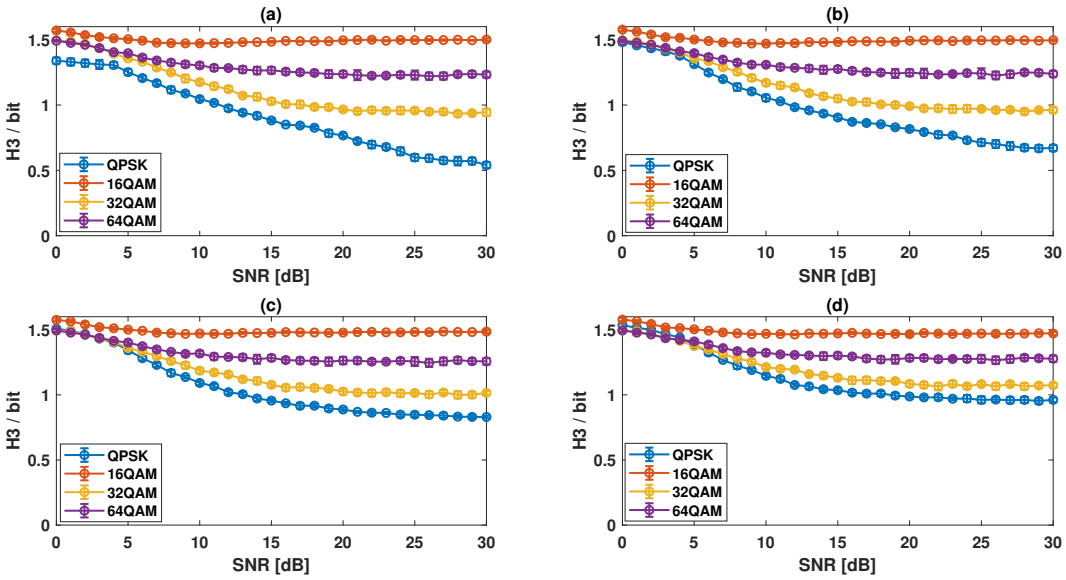

Figure 5.2: The values of H3, information entropy with 3 bins, vary with different OSNR when the residual CD is (a) 0 ps/nm, (b) 32 ps/nm, (c) 64 ps/nm, (d) 96 ps/nm in method 1.

Corresponding to the MATLAB simulation given in Chapter 3, first the different dispersion parameters are simulated. When the residual CD is set between 0 ps/nm, 32 ps/nm, 64 ps/nm, and 96 ps/nm, linewidth is 100 kHz, and baudrate is 28 GBaud, and we obtain the results shown in Fig. 5.2. It can be seen that the values of information entropy calculated on the hardware have the same trend as in MATLAB.

The information entropy will gradually increase as the CD increases. In the software, the information entropy between different modulation formats at 64 ps/nm can be well distinguished, and it can reach high accuracy at low SNR. But in hardware, the residual CD needs to be reduced to 32 ps/nm to ensure identification accuracy, because the difference between the information entropy value of QPSK and the information entropy value of 32QAM is very small when residual CD is 64 ps/nm. But even so, this method does not work well at low SNR.

The influence of the linewidth is shown in Fig. 5.3. From the figure, the test on hardware has the same result on software. The average value of information entropy and the range of fluctuations are almost unchanged. In addition, because the data transmission in the hardware takes time, it's meaningful to test the minimum number of symbols required by method 1. According to the requirement given in the design structure, three sets of data are tested here, 2048, 1024, and 512 and the test uses 64 ps/nm residual CD, 100 kHz linewidth, and 28 GBaud baudrate. The results are shown in Fig. 5.4. When the number of symbols is 1024, the average information

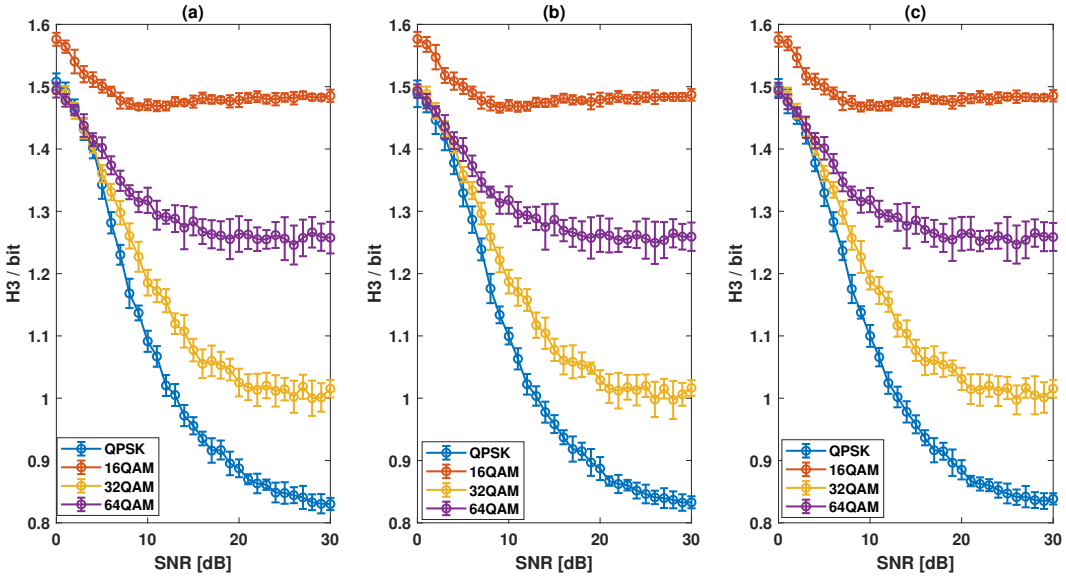

Figure 5.3: The values of H3 vary with different OSNR when the linewidth is (a)10 kHz, (b) 100 kHz, (c) 1 MHz in method 1.

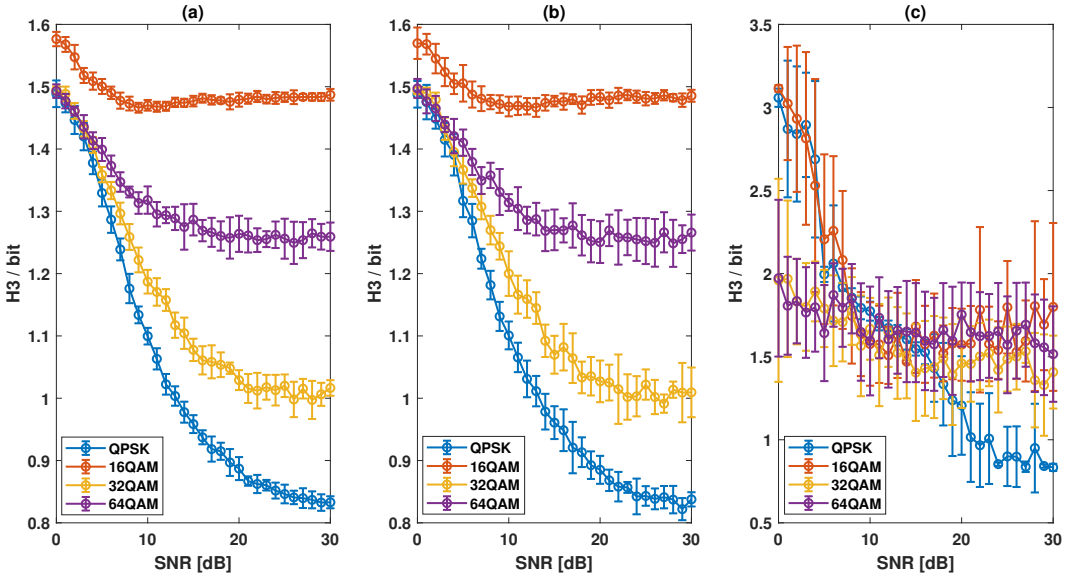

Figure 5.4: The values of H3 vary with different number of symbols in method 1. (a) 2048, (b) 1024, (c) 512.

entropy can still be maintained, but the fluctuation of the data increases. When the number of symbols is 512, information entropy can reach a value similar to other sets at high SNR. But overall, the number of symbols is too small to work properly.

5.2 Logic Simulation for Method 2

According to the discussion in Section 4.3, after the modulator is changed, the input data size can also be adjusted. We select the input size of 50, 60, 70, and 80 (as in the paper), and the parameter setting for the training set also uses Table 3.1 but the number of samples change to 20 per SNR per residual CD. The inference sets also use the same parameter setting but change the residual CD between $0 \sim 160$ ps/nm and the number of samples to 10 per SNR per residual CD. When the number of bins is 50, the accuracy of testing the inference set is around 62%, but when the number of bins is 60, 70, and 80, the accuracy can reach 73%. Therefore, we choose 60 in following simulation. After changing the size of input layer to 60, four residual

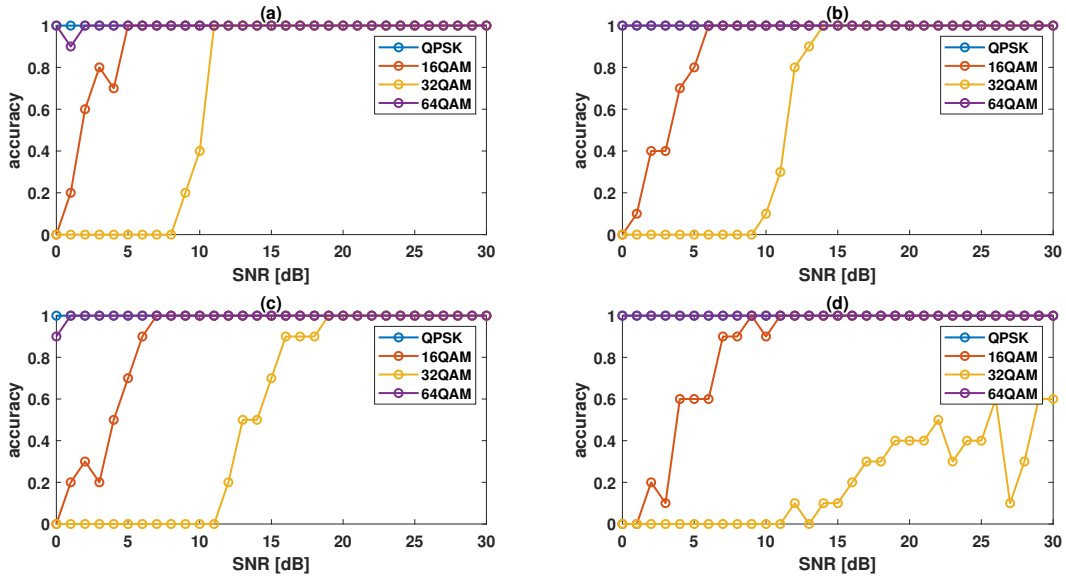

Figure 5.5: The accuracy varies with different OSNR when the residual CD is (a) 64 ps/nm, (b) 96 ps/nm, (c) 128 ps/nm, (d) 160 ps/nm in method 2.

CDs are selected to show accuracy of ANN in detail. When the linewidth is 100 kHz, and the baudrate is 28 GBaud , the residual CD is fixed at 64 ps/nm, 96 ps/nm, 128 ps/nm, and 160 ps/nm respectively in inferences sets. The test result is shown in Fig. 5.5. The simulation result shows that when the residual CD is large, the accuracy fluctuates and the performance drops significantly. This may come from the error of the activation functions, or it caused by the limitations of the network itself. In order to reduce the hardware overhead, the method selected for sigmoid is the PWL approximation that can use shifted to replace multiplication, but the average error reaches 0.014. And in ANN processing, the data has 8 decimal bits and the resolution is 0.0039, so the accuracy of the sigmoid affects the accuracy of ANN. In addition, when ANN gives the wrong identification, the simulation shows that the data input to softmax has a large difference, which should not be affected by the accuracy of softmax.

After replacing the sigmoid with the novel structure using $n \ln n$ to approximate input in Section 2.4.1, the simulation result is shown in Fig. 5.6. It can be seen that

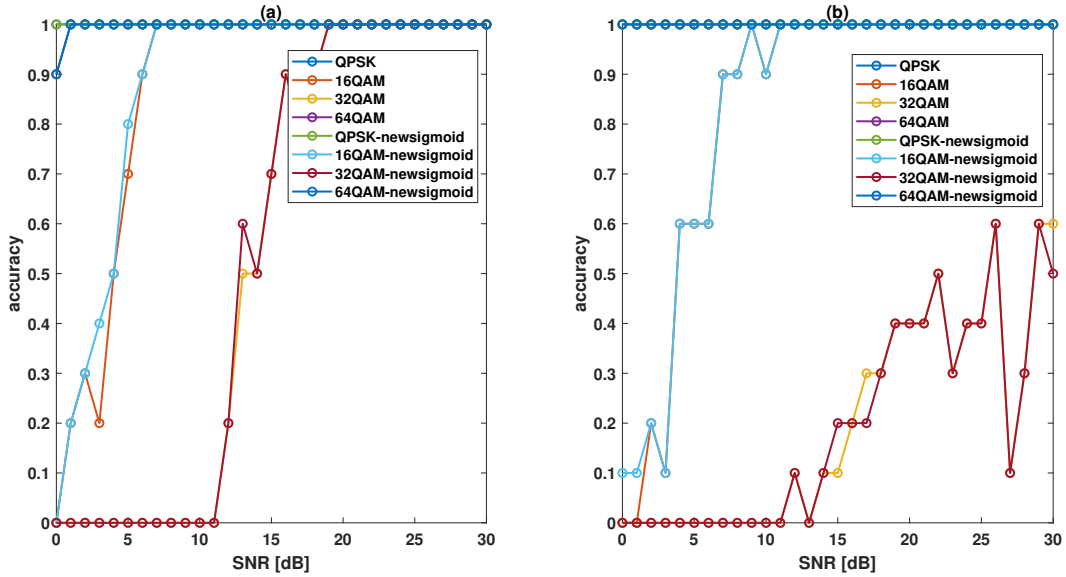

Figure 5.6: The accuracy varies with different OSNR using new sigmoid approximation when the residual CD is (a) 128 ps/nm, (b) 160 ps/nm in method 2.

when the residual CD is 128 ps/nm or 160 ps/nm, the enhanced sigmoid structure only has a small impact, so the major limitation is the network itself. The test is only performed 10 times at each SNR because of the influence of simulation time. In the following FPGA test, when inference sets have more samples, the influence of sigmoid can be seen more clearly.

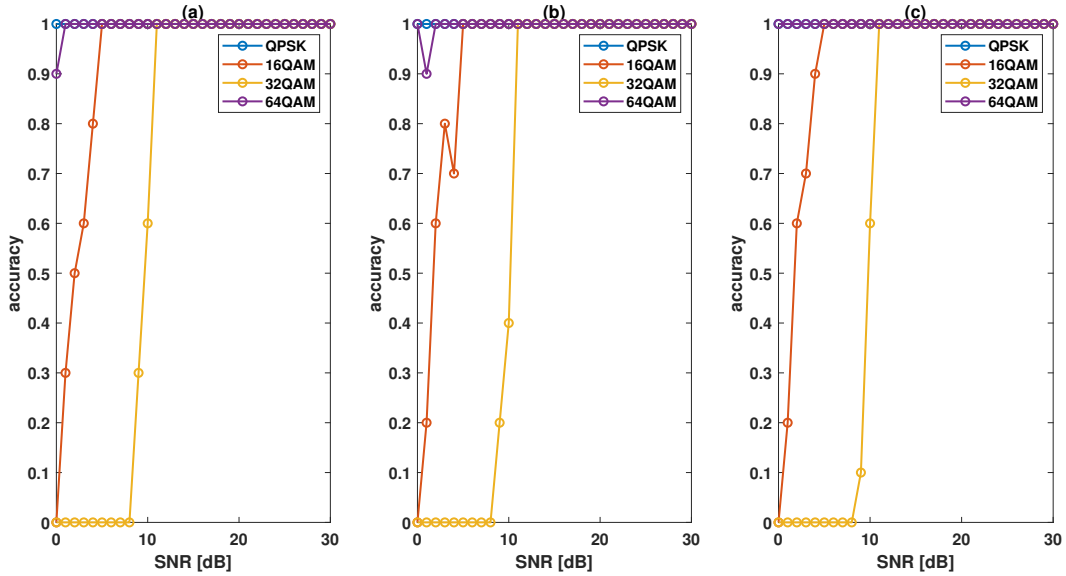

Figure 5.7: The accuracy varies with different OSNR when the linewidth is (a)10 kHz, (b) 100 kHz, (c) 1 MHz in method 2.

Similarly, the influence of linewidth can also be discussed. When the baudrate is 28

GBaud, the residual CD is 64 ps/nm, the linewidth changes from 10 kHz to 1 MHz, the test used the same ANN trained with 100 kHz linewidth for all inferences. From the results obtained in Fig. 5.7, the linewidth does not affect the accuracy of the ANN, which is consistent with the analysis in MATLAB.

Like method 1, method 2 also has requirement on the minimum number of symbols. Because the input data needs to be preprocessed and scaled between -1 and 1 before sent to ANN, this means that as long as the shape of the amplitude histogram is the same, the same ANN can be used for the different numbers of symbols. When method 1 uses amplitude histogram with a small resolution, the number of symbols can be reduced to 1024 and has the potential to be further reduced. So, when method 2 has higher resolution, the minimum number of symbols could be about 1000. In order to confirm the results, the result of the small number of symbols will be tested on the FPGA and given in the next section.

5.3 FPGA Test Results

After completing the simulation, the entire optical fiber system was transferred to an FPGA board. From the Vivado design tool, the resource utilization of the system on the VC709 FPGA board can be seen in Table 5.1. It can be seen from the results that method 1 is very small compared to other modules. According to the used method 1 structure, the processed data can be obtained after 4 clock cycles. The hardware implementation structure of method 2 selected in this project has a structure of 10 multipliers in parallel, one sigmoid and one softmax. The processing time is 96 clock cycles.

Table 5.1: Resource utilization in FPGA

Module	LUT	Register	Muxes	DSP
Transmitter and channel	9260	4903	1576	108
Method 1	328	133	0	1
Method 2	2659	1221	269	12
Method 2 with enhanced sigmoid	2797	1222	263	12

Compared to method 1, method 2 has large differences in hardware overhead and processing time. If the parallel 10 multipliers are converted into sequential processing using only multiplier, the hardware overhead would be decreased dramatically, but the processing time will increase to about 700 clock cycles. If all inputs are processed in parallel, the result can be calculated in 11 clock cycles, but the number of DSPs used is up to 600, which will greatly increase the hardware overhead. The hardware overhead of using the different sigmoid approximation is also given in Table 5.1. Although the new structure increases the use of LUTs, the change is small. The utilization of method 2 is much larger than that of method 1, but it is still small compared with the entire optical fiber system.

The FPGA test result of method 1 of different residual CDs is shown as in Fig. 5.8. The data here is obtained by testing 1000 times at each SNR point and is trans-

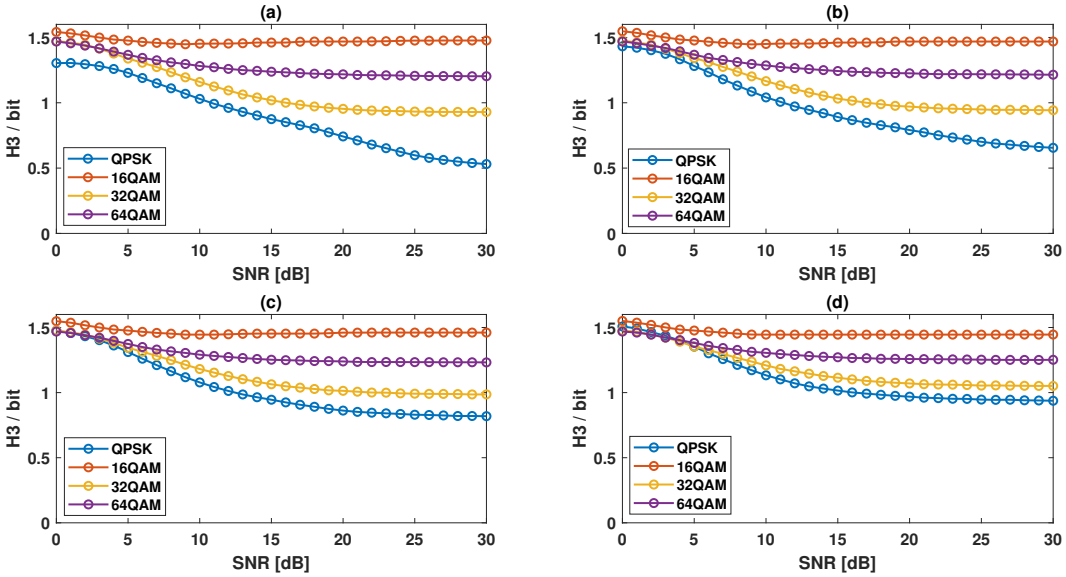

Figure 5.8: The values of H3 vary with different OSNR when the residual CD is (a) 0 ps/nm, (b) 32 ps/nm, (c) 64 ps/nm, (d) 96 ps/nm in method 1.

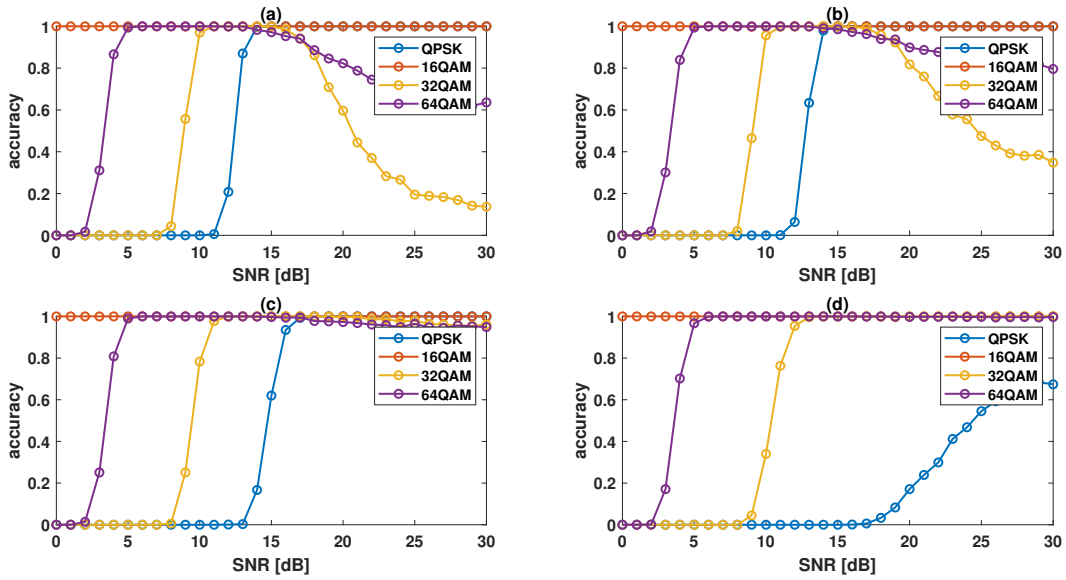

Figure 5.9: The accuracy with different OSNR when the residual CD is (a) 0 ps/nm, (b) 32 ps/nm, (c) 64 ps/nm, (d) 96 ps/nm in method 1.

mitted by UART. Because the baudrate for UART is 9600 and is much slower than the processing speed, only the average information entropy is shown in the figure. According to the average value, the corresponding threshold can be selected to perform MFI. Because the values of information entropy under high SNR corresponding to different residual CDs are different, we choose to give priority to satisfying CD as 64 ps/nm. The final result is shown in Fig. 5.9. It can be seen that when the threshold requirement of one residual CD is met, the performances corresponding to

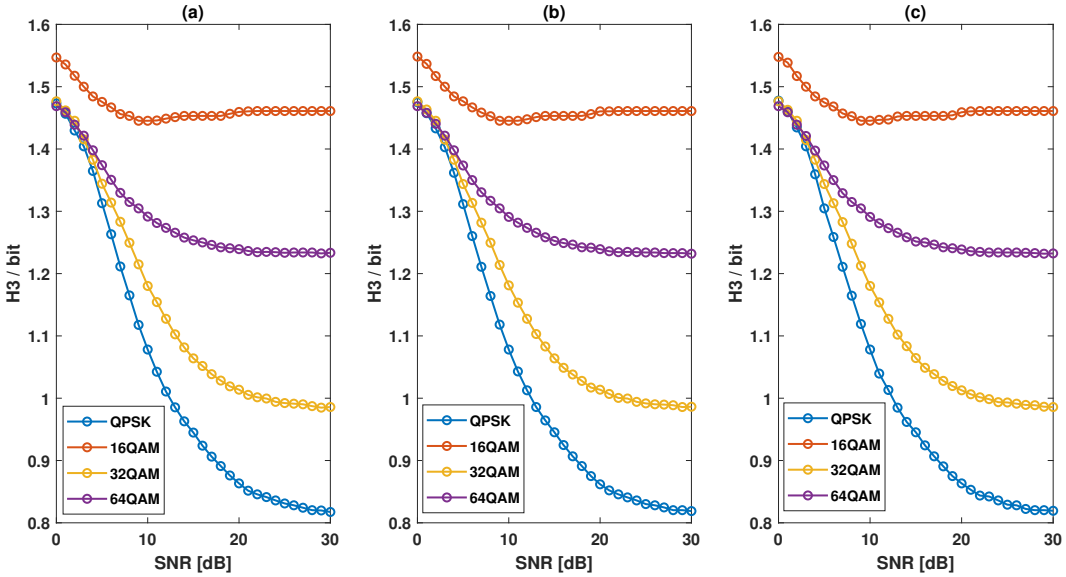

Figure 5.10: The values of H3 vary with different OSNR when the linewidth is (a) 10 kHz, (b) 100 kHz, (c) 1 MHz in method 1.

other residual CDs drop a lot. It can also be seen that although the MFI accuracy is very high, the QPSK can only be identified in high SNR. And as the residual CD increases, the SNR requirement for QPSK becomes higher.

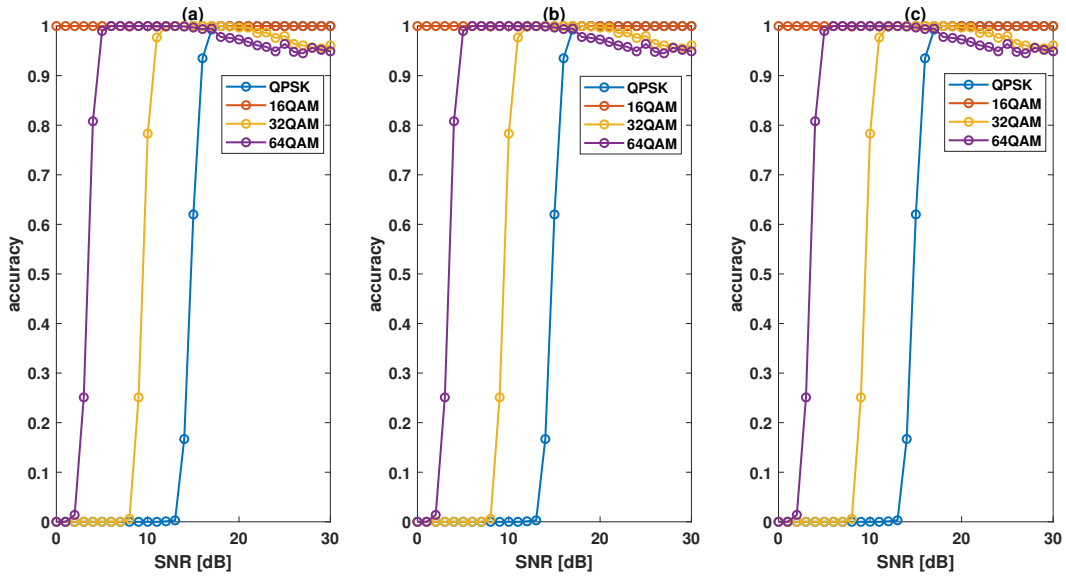

Figure 5.11: The accuracy with different OSNR when the linewidth is (a) 10 kHz, (b) 100 kHz, (c) 1 MHz in method 1.

Similarly, in Fig. 5.10, when using the same test setup as in the simulation, the average information entropy has the same result as in simulation. The accuracy in Fig. 5.11 shows the fluctuation of information entropy. It can be seen that at high

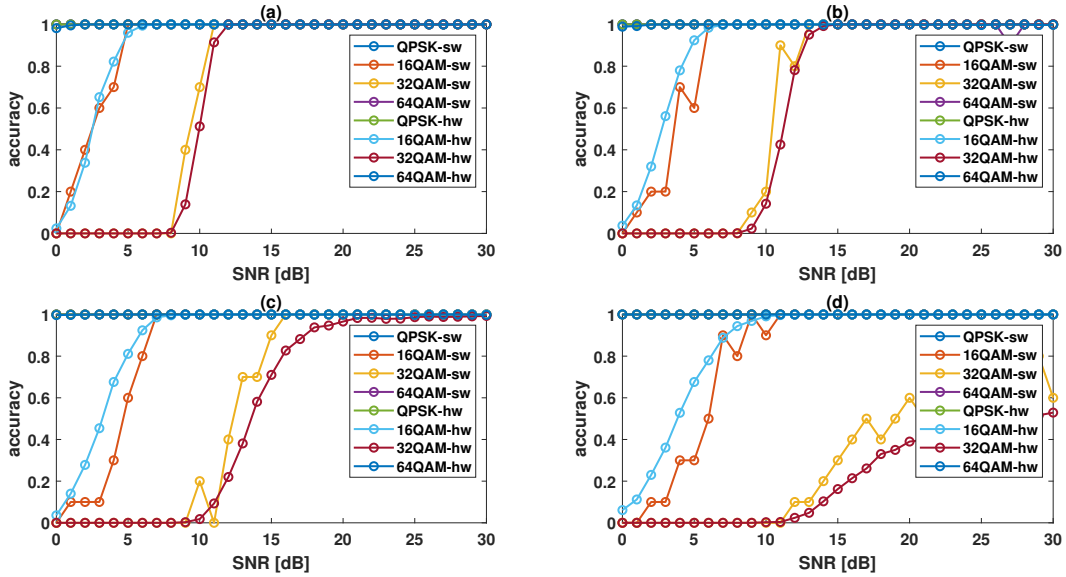

Figure 5.12: The accuracy of MATLAB and FPGA test varies with different OSNR when the residual CD is (a) 64 ps/nm, (b) 96 ps/nm, (c) 128 ps/nm, (d) 160 ps/nm in method 2.

SNR, the accuracy of 32QAM identification decreases. This is because the choice of threshold between 32QAM and QPSK will affect the lowest SNR that QPSK can be identified and the accuracy of identifying 32QAM at high SNR. And although the FPGA test has chosen a low threshold to sacrifice more QPSK, the accuracy still be decrease at high SNR for 32QAM due to data fluctuations.

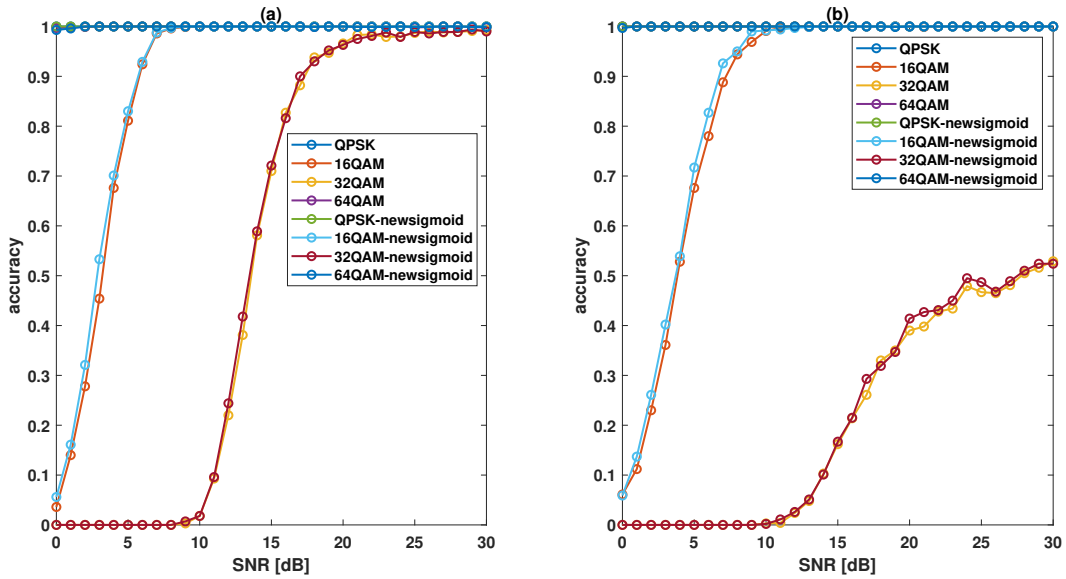

Figure 5.13: The accuracy of using different sigmoid approximations varies with different OSNR using new sigmoid approximation when the residual CD is (a) 128 ps/nm, (b) 160 ps/nm in method 2.

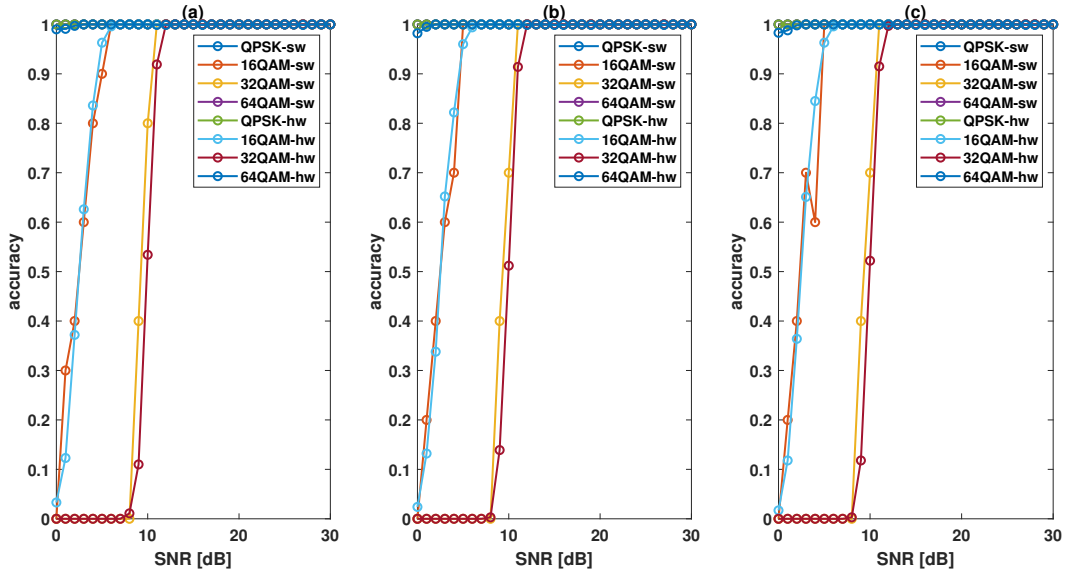

Figure 5.14: The accuracy of MATLAB and FPGA varies with different OSNR when the linewidth is (a)10 kHz, (b) 100 kHz, (c) 1 MHz in method 2.

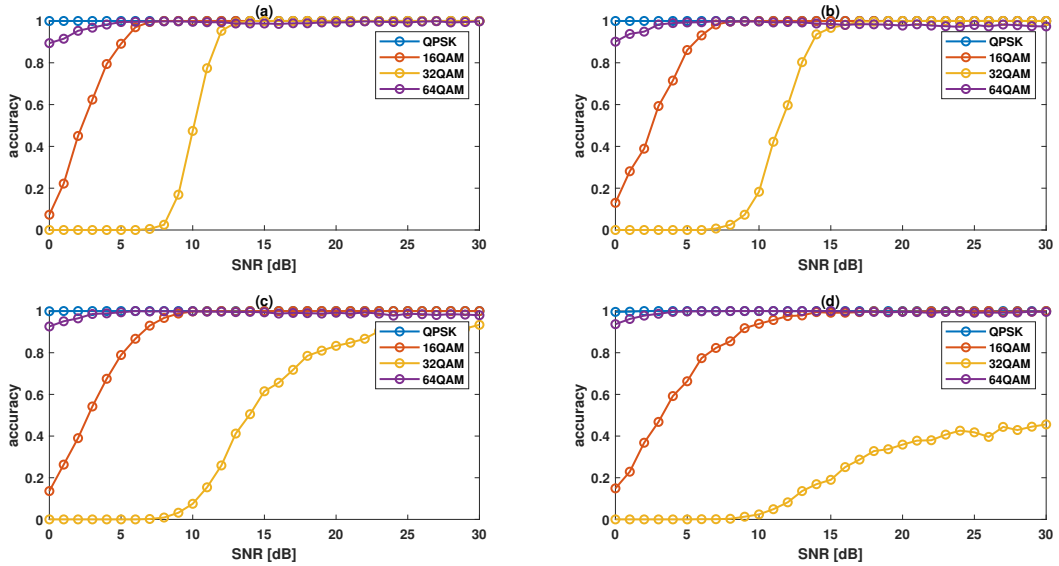

Figure 5.15: The accuracy varies with different OSNR when the number of symbols changes to 1000 in method 2.

The FPGA test results of method 2 with the residual CD variation corresponding to the previous simulation results are shown in Fig. 5.12. Here, 1000 inferences have been performed for each SNR of each format. At the same time, Fig. 5.12 also shows using AH obtained by simulation to test the same ANN on MATLAB, and each SNR was tested 10 times. From Fig. 5.12, the performance loss caused by the hardware on the fixed data length and the approximation of the activation function is very small on the low residual CD. On the high residual CD, there are more

obvious changes. The impact of low-precision sigmoid approximation can be seen more clearly in FPGA test. Here, when residual CD are 128 ps/nm and 160 ps/nm, the result of using a high-precision sigmoid approximation structure is shown in Fig. 5.13. It can be seen that the accuracy has only improved a bit. Comparing the 100% accuracy achieved by the software in Fig. 5.12 and the 98% accuracy at high SNR in Fig. 5.13, it can be seen that the fixed data length still has a certain impact on the accuracy. In addition, the test results of linewidth are also displayed by comparing with the results of MATLAB. Fig. 5.14 shows that the results of the two test are basically coincident.

As mentioned earlier, method 2 may accept the smallest number of symbols as small as 1000. In FPGA test, we can multiply the amplitude histogram generated by 1000 symbols by 2.5 and regard this as the result of 2500 symbols. If the amplitude histogram from 1000 symbols has the same shape as using 2500 symbols, the ANN trained with 2500 symbols can be used normally. The final result is shown in Fig. 5.15. It can be seen that in addition to QPSK, the accuracy of other modulation format has a certain decline. But in general, when the residual CD is small, using 1000 symbols can still bring good results.

6

Conclusion

In this thesis, we first discussed three methods to perform MFI on a signal whose modulation format can vary between QPSK, 16QAM, 32QAM, and 64QAM. And for each method, the effects of different impairments are individually analysed using MATLAB. After discussing the reliability of those methods in MATLAB, based on the approximate hardware overhead required to implement different methods, we selected two methods.

The first one is the conventional method based on calculating information entropy. This method has a small hardware overhead and a short processing time. But this method can only be applied to fixed residual CDs or residual CDs with a small variation range. Also, due to the limitations of the method itself, it cannot achieve the minimum SNR requirement to identify QPSK. And at a higher residual CD, the accuracy at high SNR will drop to 98%. This performance loss is caused by the limitation of the method itself, which cannot be improved by adjusting the resolution or the number of analysed symbols in the hardware processing.

The second method is to use an ANN to identify AH. Compared with the first method, this method requires more hardware resources and longer processing time. But this method can tolerate a higher CD. When the residual CD is 128 ps/nm, the accuracy within the target SNR range is at least 99%, and this accuracy can be further improved by increasing the resolution in calculation. When the residual CD is 180 ps/nm, the accuracy of the ANN will be limited due to the method itself. In addition, method 1 needs to set different thresholds for different residual CDs, but method 2 can realize using one ANN for large range of CDs by given the suitable training set.

This thesis has given two different methods for real-time MFI implementation in the fiber optical system on FPGA, and a large number of tests have proved the reliability of these two methods. After knowing the possible modulation formats and channel conditions of the optical fiber system, we can decide which method should be used. Method 1 has low hardware overhead but high channel requirements and the method 2 has higher hardware overhead but high impairments tolerance.

Future work may include using these two methods to test more modulation formats, such as 256 QAM. From the current point of view, the principle of method 1 can detect more formats. But in the same way, the requirements for channel quality may be further improved. For method 2, because the AH of 256 QAM and other modulation formats are still quite different, when training the ANN, if additional training data is added, the realized ANN is likely to reach the expected goal. But

also, because the AHs of the high modulation formats like 64 QAM and 256 QAM are greatly affected by residual CD, the channel requirements of the second method may also be increased. In addition, the method of this project to achieve residual CD is to directly generate CD according to the short fiber length. But the actual system usually uses long fiber and CD compensation. So the influence of residual CD may change in the real system. The difference can be further discussed and used to test for two methods.

Bibliography

[1] V. GLópez and L. Velasco, *Elastic Optical Networks Architectures, Technologies, and Control*. Springer, 2016.

[2] M. C. Tan, F. N. Khan, W. H. Al-Arashi, Y. Zhou, and A. P. T. Lau, "Simultaneous optical performance monitoring and modulation format/bit-rate identification using principal component analysis," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 6, no. 5, pp. 441–448, 2014.

[3] Z. Zhao, A. Yang, and P. Guo, "A modulation format identification method based on information entropy analysis of received optical communication signal," *IEEE Access*, vol. 7, pp. 41 492–41 497, 2019.

[4] G. Keiser, *Optical Fiber Communications*. McGraw-Hill Companies, 2011.

[5] G. Liu, K. Zhang, R. Proietti, H. Lu, Z. Ding, and S. J. Ben Yoo, "First demonstration of fpga-based hitless flexible receiver with blind modulation format identification," in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, 2018, pp. 1–3.

[6] D. Kouznetsov, J.-F. Bisson, K. Takaichi, and K. ichi Ueda, "High-power single-mode solid-state laser with a short, wide unstable cavity," *J. Opt. Soc. Am. B*, vol. 22, no. 8, pp. 1605–1619, Aug 2005.

[7] R. Loudon, *The quantum theory of light*. OUP Oxford, 2000.

[8] G. P. Agrawal, *Fiber-Optic Communication Systems*. Wiley, 2010.

[9] L. Jiang, L. Yan, A. Yi, Y. Pan, M. Hao, W. Pan, and B. Luo, "Blind modulation format identification based on fourier fitting for coherent receivers," in *2018 23rd Opto-Electronics and Communications Conference (OECC)*, 2018, pp. 1–2.

[10] J. Zhang, M. Gao, W. Chen, Y. Ye, Y. Ma, Y. Yan, H. Ren, and G. Shen, "Blind and noise-tolerant modulation format identification," *IEEE Photonics Technology Letters*, vol. 30, no. 21, pp. 1850–1853, 2018.

[11] F. N. Khan, K. Zhong, W. H. Al-Arashi, C. Yu, C. Lu, and A. P. T. Lau, "Modulation format identification in coherent receivers using deep machine learning," *IEEE Photonics Technology Letters*, vol. 28, no. 17, pp. 1886–1889, 2016.

[12] Y. Zeng, M. Zhang, F. Han, Y. Gong, and J. Zhang, "Spectrum analysis and convolutional neural network for automatic modulation recognition," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 929–932, 2019.

[13] J. W. Kim and C. H. Lee, "Modulation format identification for square M-QAM signals by using a neural network," *Optics InfoBase Conference Papers*, vol. Part F83-ACPC 2017, pp. 18–20, 2017.

[14] M. Kulin, T. Kazaz, I. Moerman, and E. De Poorter, "End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications," *IEEE Access*, vol. 6, pp. 18 484–18 501, 2018.

[15] A. O. Abdul Salam, R. E. Sheriff, S. R. Al-Araji, K. Mezher, and Q. Nasir, "A unified practical approach to modulation classification in cognitive radio using likelihood-based techniques," in *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2015, pp. 1024–1029.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[17] D. Wang, M. Zhang, J. Li, Z. Li, J. Li, C. Song, and X. Chen, "Intelligent constellation diagram analyzer using convolutional neural network-based deep learning," *Opt. Express*, vol. 25, no. 15, pp. 17 150–17 166, Jul 2017.

[18] F. Temurtas, A. Gulbag, and N. Yumusak, "A study on neural networks using Taylor series expansion of sigmoid activation function," in *Computational Science and Its Applications – ICCSA 2004*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 389–397.

[19] I. Tsmots, O. Skorokhoda, and V. Rabyk, "Hardware implementation of sigmoid activation functions using FPGA," in *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, 2019, pp. 34–38.

[20] K. Leboeuf, A. H. Namin, R. Muscedere, H. Wu, and M. Ahmadi, "High speed VLSI implementation of the hyperbolic tangent sigmoid function," in *2008 Third International Conference on Convergence and Hybrid Information Technology*, vol. 1, 2008, pp. 1070–1073.

[21] R. W. Duren, R. J. Marks II, P. D. Reynolds, and M. L. Trumbo, "Real-time neural network inversion on the SRC-6e reconfigurable computer," *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 889–901, 2007.

[22] Z. Qin, Y. Qiu, H. Sun, Z. Lu, Z. Wang, Q. Shen, and H. Pan, "A novel approximation methodology and its efficient VLSI implementation for the sigmoid function," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 12, pp. 3422–3426, 2020.

[23] Y. Gao, W. Liu, and F. Lombardi, "Design and implementation of an approximate softmax layer for deep neural networks," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5.

[24] X. Geng, J. Lin, B. Zhao, A. Kong, M. M. S. Aly, and V. Chandrasekhar, "Hardware-aware softmax approximation for deep neural networks," in *Computer Vision – ACCV 2018*, C. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Cham: Springer International Publishing, 2019, pp. 107–122.

[25] X. Dong, X. Zhu, and D. Ma, "Hardware implementation of softmax function based on piecewise LUT," in *2019 IEEE International Workshop on Future Computing (IWOFC)*, 2019, pp. 1–3.

[26] D. Zhu, S. Lu, M. Wang, J. Lin, and Z. Wang, "Efficient precision-adjustable architecture for softmax function in deep learning," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 12, pp. 3382–3386, 2020.

[27] E. Börjeson, C. Fougstedt, and P. Larsson-Edefors, "Towards FPGA emulation of fiber-optic channels for deep-BER evaluation of DSP implementations," in *OSA Advanced Photonics Congress (AP) 2019 (IPR, Networks, NOMA, SPPCom, PVLED)*. Optical Society of America, 2019, p. SpTh1E.4.

[28] M. Khafaji, H. Gustat, F. Ellinger, and C. Scheytt, "General time-domain representation of chromatic dispersion in single-mode fibers," *IEEE Photonics Technology Letters*, vol. 22, no. 5, pp. 314–316, 2010.

[29] R. Lyons, *Understanding Digital Signal Processing*. Pearson, 2011.