A Novel CNN-Based FSK Demodulator With Efficient FPGA Implementation

AmirHossein Sadough
Department of Electrical Engineering
Shahid Rajaee Teacher Training University
Tehran, Iran
amirhosseinsadough@gmail.com

Sina Rezaeeahvanouee
Department of Electrical and Computer Engineering
University of Minnesota
Minneapolis, US
rezae031@umn.edu

2023 31st International Conference on Electrical Engineering (ICEE) | 979-8-3503-1256-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICEE59167.2023.10334792

Abstract— **Nowadays, neural networks have become a new approach to achieving the destinations of researchers due to their accuracy. Neural networks are widely used in machine vision, image classification, and sound detection, but in the field of signal processing, it still has no content of the extent in the field of image processing. This is not due to the inability of deep learning to meet the needs of signal processing, but because that is an ultra-modern solution. This paper proposed a novel CNN-Based FSK demodulator (CFD) that uses a compact convolutional neural network as a special method to demodulate the FSK signal. The proposed CNN network has a minimal computational load and is optimized in terms of hardware implementation complexity. The designed CNN network has three layers: a convolutional layer, a Max-Pooling layer, and a fully connected layer. The CFD is designed to demodulate an FSK signal of 20±1MHz with a bit rate of 1Mbps. The BER of lower than 10^{-6} has been achieved at E_b/N_0=15dB. The resource utilization of implemented proposed demodulator on FPGA only shows 970 and 751, for LUT and FF respectively, which is minimal and indicates the efficiency of our method.**

Keywords—FSK, CNN, Demodulator, CFD, FPGA, BER, Implementation

I. INTRODUCTION

The accuracy feature of the neural networks, particularly the Convolutional Neural Networks (CNN) in image classification and detection [1], [2], encourages attention to employ these networks in signal processing. This leads to a reform of signal processing algorithms based on deep learning to enhance performance and optimize the design [3].

As a way to meet energy efficiency and convenience in wireless communication, frequency shift keying (FSK) modulation is one of the most efficient methods. The FSK modulation technology has been used in a variety of applications including biomedical implants [5] and [6], and low-power devices [7]-[11]. The characteristics of this modulation approve the demands of low power. On the other hand, efforts to use different methods to implement FSK modulation to obtain a minimum of energy consumption and the area, establish an attractive context for researchers yet. Based on FSK modulation, in bits 0 and 1, the modulator transmits frequencies F1 and F2 respectively. In recent years, vast techniques have been used to develop FSK demodulators. A conventional method of demodulating the FSK signal involves leveraging a mixer and an envelope detector. The PLL-based FSK demodulator is an alternative method offered in [12]. In [13] a novel technique has been exploited that benefit a neural network to demodulate noncoherent modulated signals. Furthermore, neural networks have been used for signal demodulation in [14], where a complex noncoherent neural network is has exploited to demodulate the OOK signal. Since digital systems implementation requires an appropriate implementation platform, Field Programmable Gate Array (FPGA) is an adaptable case. Due to the capability of reconfiguration, parallelism, digital signal processing elements, and flexibility, FPGAs are considerably used to apply signal processing algorithms, neural networks, and any digital system. Moreover, their energy consumption, efficiency, and performance have made a significant contribution to integrating FPGA into systems.

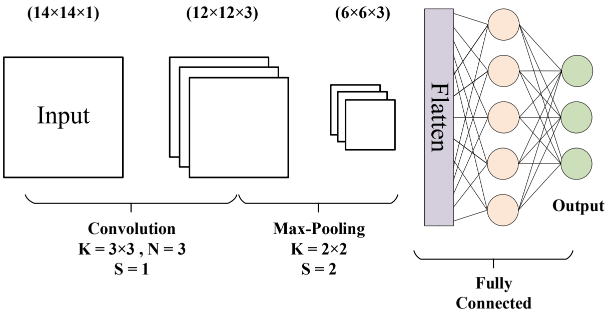


Fig. 1. An example of CNN networks.

In this paper, we proposed a novel convolutional neural network-based FSK demodulator which implemented on FPGA. The main contribution of this paper will be as follow:

- Design of a compact CNN network for demodulating FSK signal and data extraction.
- Optimum implementation of proposed demodulator on FPGA.

This paper is organized as follows: Section II introduces briefly CNN networks and their architecture. Section III describes the proposed FSK demodulator. Section IV focused on demonstrating the performance analysis and implementation results. The conclusion of this paper is provided in Section V.

II. CNN NETWORKS

A. Architecture

An example of a convolutional neural network is depicted in Fig. 1 which has three layers including a convolutional neural network, a Max-Pooling layer, and a Fully-Connected layer. CNN networks have numerous types of layers. These

979-8-3503-1256-0/23/$31.00 2023 IEEE

TABLE I.
FSK DEMODULATOR KEY SPECIFICATION

Parameters	Value
$f_{Carrier}$(MHz)	20 [a]
$f_{BitRate}$(Mbps)	1
$f_{Deviation}$(MHz)	1
f_s(MHz)	8

[a] Tunable

layers contain convolutional layer, Max-Pooling layer, and Fully-Connected layer. In the convolutional layer, features are extracted by convolving a set of filters on input. A Two-dimensional convolutional layer is calculated as follows:

$$O_{y,x} = \sum_{h=0}^{k_h} \sum_{w=0}^{k_w} I_{y+h,x+w} \times K_{h,w} \tag{1}$$

Where I is the input of the convolutional layer, K is the filter kernel, and O is the output of the layer. The Max-Pooling layer is used to diminish data density between layers. A mask with determined window size and stride, slips on the output of the previous layer and obtains the maximum of pixels in the mask. For instance, a 2×2 Max-Pooling on input with a dimension of 8×8 generates the output with a dimension of 4×4. To conclude, the Fully-Connected layer generally applies at the end of the neural networks, which will be defined as follows:

$$Y_j = \sum_{i=0}^{n} X_i \times W_{i,j} + B_j \tag{2}$$

Where X is the input of Fully-Connected layer, W is the weight, B is the weight bias, and Y is the output. The classification layer is the output of the last Fully-Connected layer. The number of members for this layer is equal to desired classes that are considered for conclusion by the network.

B. Exploiting

The deep learning network operations are summarized in two steps: First, network training. The training process uses known inputs for network training and in other words, this means that the network weight correction is such that it produces the desired output for the known input [1]. To achieve that, a bunch of data with known labels applies to the neural network and the network concludes for each input. By comparing the obtained result and the expected result, the inference error is acquired for that input. Subsequently, the network in the back-propagation phase, taking into account the loss function, updates the weight of the network. Secondly, the Inference process. After training to achieve minimal error in the conclusion, the inference phase of the network will be accessible. where it may be used for unknown inputs of the classes taught to the network. In other words, weights are prepared to utilize in the desired application.

III. CNN-BASED FSK DEMODULATOR

A. Structure Elucidation

The signal specifications of the proposed FSK demodulator are mentioned in Table I. Since the sampling rate f_s = 8MHz and the bitrate $f_{BitRate}$ = 1Mbps have been availed, the number of samples per symbol will be 8 samples. In addition, the deviation frequency f_{dev}= ±1MHz with respect to the baseband is designated due to attaining proper selectivity ability. The frequency spectrum of the received FSK signal at the proposed demodulator input is illustrated in Fig. 2. Similarly, Fig. 3 shows the In-phase and quadrature components of desired FSK signal at the baseband. For such an FSK signal, a Convolutional neural network-based FSK Demodulator (CFD) has been proposed which is indicated in Fig. 4. This modulator is composed of two main blocks: First, a packing block that reforms In-phase and quadrature components of the received FSK signal to a 2-D array. Second, a CFD processor block that demodulates the FSK signal. The CFD processor block is based on a compact CNN network, where CNN is responsible to detect 0 or 1 data. The proposed CNN network shown in Fig. 5 is constructed by a

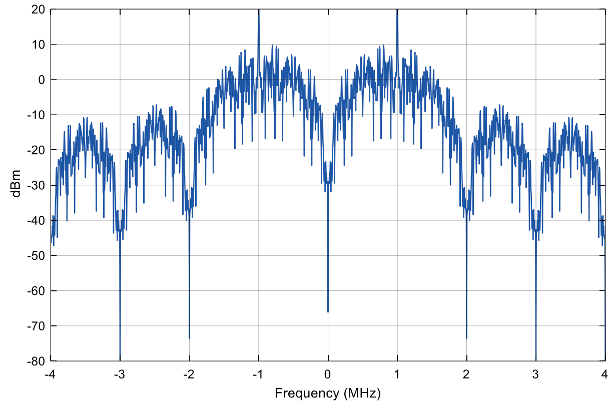


Fig. 2. Spectrum of the FSK signal in baseband.

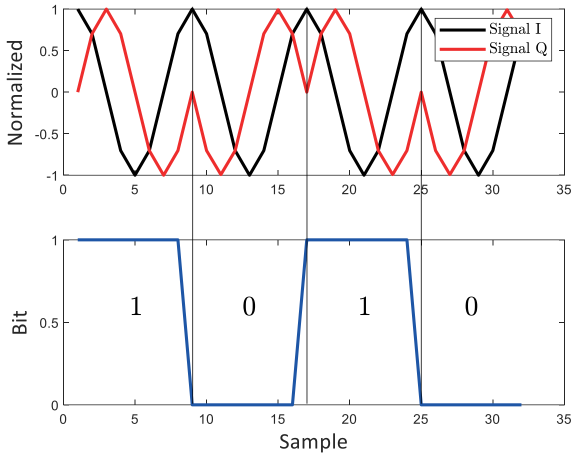


Fig. 3. Baseband FSK signal and corresponding 0/1 data depicted from top to bottom.

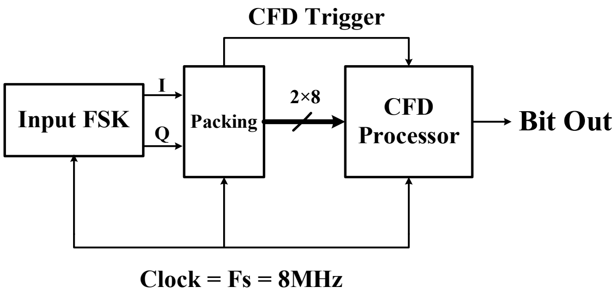


Fig. 4. CNN-Based FSK demodulator architecture

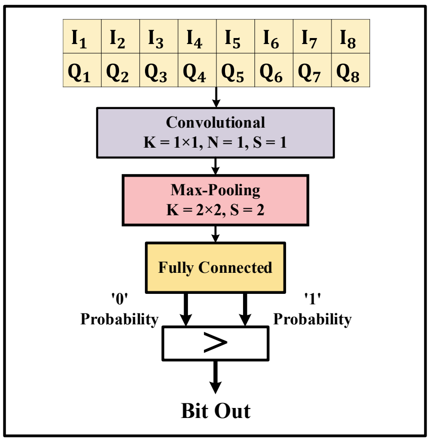
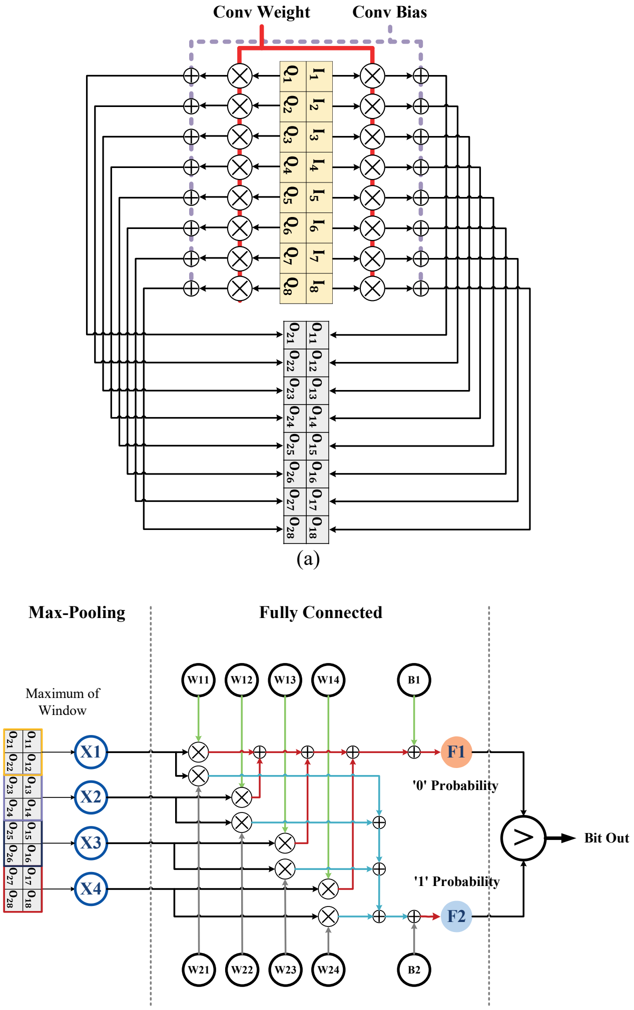


Fig. 5. The proposed convolutional neural network for data classification.

TABLE II.
LEARNING CONFIGURATIONS

Parameters	Value
Optimizer	Nadam
Learning Rate	0.001
Beta_1	0.9
Beta_2	0.999
Epsilon	1e^{-7}

convolutional layer that has just one filter with a dimension of 1×1 and stride = 1. Therefore, one weight and one bias are the total trainable parameters of this layer. In the following, the Max-Pooling layer has been appointed with a mask dimension of 2×2 with stride = 1. In the last part of the network structure, a Fully-Connected layer with two output neurons specifies the probability to be data 0 or 1 for the received symbol. Since this layer has 4 inputs and 2 outputs, to that account the number of learning parameters will be 10. Altogether, the proposed CNN has just 12 parameters and this advantage produce conceivable to implement it in a minimum area. The proposed neural network is detailed in Fig. 5. A 2D input array as the input of the network has a size of M×N, that M is equal to 2 since the input signal is represented by both I and Q components for each sample. On other hand, $N = f_s/f_{BitRate}$ has been considered which means N will be equal to 8. Consequently, the input array of the designed CNN has a dimension of 2×8. For training the proposed CNN, Datasets of 0 and 1 symbols have been collected that are affected by AWGN channel with a diversity of E_b/N_0. The variety amount of E_b/N_0 is from 0dB to 25dB to cover all situations, it results in precise adjustment of network weights and ultimately the ability to operate under different noise conditions. Thus, 2e^5 symbols have been used for training and 5e^4 symbols for validation in the network training process. The proposed CNN has been trained by TensorFlow.Keras library in PyTorch. To become clear, Table II reveals the learning configuration. It is straightforward that the network parameters are normally represented as float32. So, quantization is crucial to represent parameters in a fixed-point format, which enables the implementation of structure on FPGA. The quantization of the network parameters has been done by a 5-bit signed (FP5) representation that ensures classification accuracy. According to Fig. 4, demodulation operates via CFD Processor. The packing block gathers data in an array format of 2×8 and feeds the CFD processor. Whenever the array completes with 8 unprecedented pairs of In-phase and quadrature samples, the CFD Trigger is produced from the packing block. The CFD Processor block receives the package by the packing block and performs the network inference when CFD Trigger is valid. '0' probability or '1' probability obtains as two classes in the output of Fully-Connected and then comparing between these two probabilities reveals the input symbol corresponds to which data.



(a)

(b)

Fig. 6. The implementation block diagram of proposed CNN network on FPGA: a) Convolutional and b) Max-Pooling and fully-connected layer.

B. FPGA Implementation

Fig. 6 shows the implementation block diagram of the CFD processor on FPGA. Contrary to the fact that CNN networks are often complicated in implementation, this complexity is not seen in the proposed architecture. The convolutional layer has more complexity than other layers because most networks in the convolutional layer have a 3D convolution operation with numerous filters. Unlike conventional CNN networks, the proposed CNN has just one filter of one dimensional in the convolutional layer. Thus, (1) can be corrected for CFD convolutional layer as follows:

$$O_{y,x} = I_{y,x} \times K \tag{3}$$

With these explanations, CFD has not any complexity in the implementation phase. As well, since the network parameters have been characterized in FP5 format, the multiplication

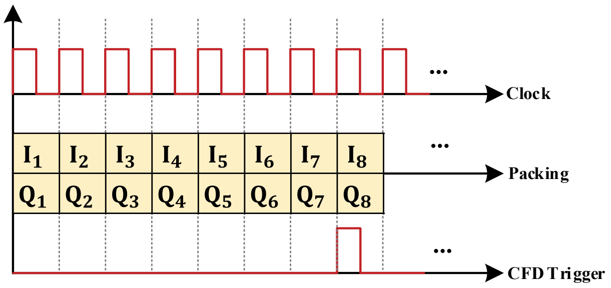


Fig. 7. Packing and CFD Trigger versus clock.

operations can be synthesized by LUTs and it has no compulsion to implement by DSP blocks. Accordingly, the proposed structure does not put a redundant burden on hardware resources.

Timing diagram of the Completion of collecting samples and packaging a symbol by the packing block alongside creating a CFD Trigger with respect to the clock is shown in Fig. 7. CFD Trigger timing period is equal to 8, which guarantees to collect of all samples of a new symbol. On the side of the CFD processor, the calculation latency is equal to 6 which means that is lower than CFD Trigger and confirms the demodulation process continues unceasingly.

IV. Implementation Results and Analysis

A. Implementation Resource Utilization

The implementation of the CFD on the FPGA has been done using VHDL and in the XILINX VIVADO development environment. The target FPGA device that has been selected to evaluate the amount of resource utilization is Xilinx XC7A50T. hardware resource utilization is one of the imperative criteria for evaluating the design and demonstrating achieved optimization. The implementation result of the proposed CFD on FPGA in Table III is declared. Only has been consumed 970 and 751 for LUT and FF respectively, which is 3% and 1% of the total accessible LUT and FF, respectively for the target device. These values prove the proposed FSK demodulator is widely optimum in the aspect of resource utilization.

B. BER Performance

One of the most important criteria for evaluating demodulators is their performance at the bit error rate. For this purpose, a quantity of 10^6 from FSK symbols for each E_b/N_0 from 0db to 15dB with 1dB resolution step under AWGN channel has been generated by MATLAB software as test samples. The performance of the CFD demodulator has been evaluated using these samples and the BER has been extracted. To achieve this, collected samples have been used as input in hardware simulation using the Xilinx ISIM environment, and demodulated data has been extracted. The comparison result of obtained BER by the extracted data from hardware simulation for the CFD demodulator and the Ideal-noncoherent FSK demodulator is shown in Figure. 8. According to BER curve, in E_b/N_0 less than 11dB, obviously CFD performance is better than the Ideal-noncoherent FSK demodulator which indicates the potency of the proposed CFD to demodulate symbols that have been severely demolished in the AWGN channel. On the other hand, BER equal to 10^{-6} indicates excellent quality of communication. The CFD demodulator has achieved this quality approximately at $E_b/N_0 = 15$dB, which has not much different from the ideal-noncoherent-FSK demodulator.

Table III.
Resource Utilization

	CNN-Based FSK Demodulator
Device	XC7A50T
LUT	970 (3%)
FF	751(1%)

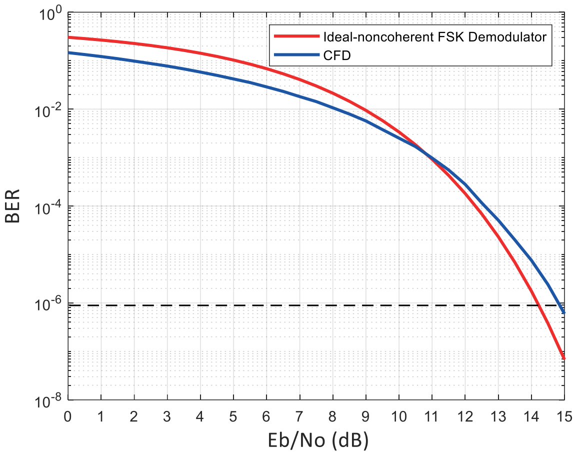


Fig. 8. The BER curve of Ideal-Noncoherent FSK demodulator and CFD.

V. Conclusion

A Novel CNN-Based FSK Demodulator has been proposed and implemented on FPGA which utilizes 970 and 751 for LUT and FF, respectively. In addition, proper BER performance in comparison to an Ideal-noncoherent FSK demodulator shows the design is optimized for use in low-power applications and meets the requirements of each application with a limitation of the use of hardware resources.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, May 2015.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[3] Nguyen, Hai N., and Guevara Noubir. "Towards an AI-Driven Universal Anti-Jamming Solution with Convolutional Interference Cancellation Network." *arXiv preprint arXiv:2203.09717* (2022).

[4] Y. Hwang, B. Hwang, H. Lin, and J. Chen, ‘PLL-based contactless energy transfer analog FSK demodulator using high-efficienc rectifier’, IEEE Transactions on Industrial Electronics, Year: 2013, Volume: 60, Issue: 1, pp. 280-290.

[5] M. Ghovanloo and K. Najafi, “A wideband frequency-shift keying wireless link for inductively powered biomedical implants,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 12, pp. 2374–2383, Dec. 2004.

[6] C. Sauer, M. Stanacevic, G. Cauwenberghs, and N. Thakor, “Power harvesting and telemetry in CMOS for implanted devices,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 12, pp. 2605–2613, Dec. 2005.

[7] S. Hu *et al.*, “A type-II phase-tracking receiver,” *IEEE J. Solid-State Circuits*, early access, Jul. 17, 2020, doi: 10.1109/JSSC.2020.3005797.

[8] M. Tamura *et al.*, “A 0.5V BLE transceiver with a 1.9mW RX achieving −96.4dBm sensitivity and 4.1dB adjacent channel rejection

at 1MHz offset in 22nm FDSOI," in *ISSCC Dig. Tech. Papers*, Feb. 2020, pp. 468–470.

[9] M. Ding *et al.*, "A 0.8V 0.8mm2 Bluetooth 5/BLE digital-intensive transceiver with a 2.3mW phase-tracking RX utilizing a hybrid loop filter for interference resilience in 40nm CMOS," in *ISSCC Dig. Tech. Papers*, Feb. 2018, pp. 446–448.

[10] H. Liu *et al.*, "An ADPLL-centric Bluetooth low-energy transceiver with 2.3mW interference-tolerant hybrid-loop receiver and 2.9mW singlepoint polar transmitter in 65nm CMOS," in *ISSCC Dig. Tech. Papers*, Feb. 2018, pp. 444–446.

[11] Y. Liu *et al.*, "A 1.2 nJ/bit 2.4 GHz receiver with a sliding-IF phaseto-digital converter for wireless personal/body area networks," in *ISSCC Dig. Tech. Papers*, Feb. 2014, pp. 166–167.

[12] S. Delshadpour and M. Geng, "A PLL Based FSK Demodulator With Auxiliary Path," *2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS)*, 2020, pp. 1-4, doi: 10.1109/LASCAS45839.2020.9069002.

[13] P. E. Gorday, N. Erdöl and H. Zhuang, "Complex-Valued Neural Networks for Noncoherent Demodulation," in *IEEE Open Journal of the Communications Society*, vol. 1, pp. 217-225, 2020, doi: 10.1109/OJCOMS.2020.2970688.

[14] P. Gorday, N. Erdöl and H. Zhuang, "A Noncoherent Incremental Learning Demodulator," *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2020, pp. 0200-0206, doi: 10.1109/UEMCON51285.2020.9298172.