This document is a preprint of: Z. Li, A. Nimr and G. Fettweis, "Implementation and Performance Measurement of Flexible Radix-2 GFDM Modem," in Proceedings of IEEE 5G World Forum (WF-5G 2019), Dresden, Germany, (pp. 130-134), Sep 2019. DOI:10.1109/5GWF.2019.8911718

Implementation and Performance Measurement of Flexible Radix-2 GFDM Modem

Zhongju Li†, Ahmad Nimr*, Gerhard Fettweis*
†Barkhausen Institut, Dresden, Germany
*Vodafone Chair Mobile Communication Systems, Technische Universität Dresden, Germany
†zhongju.li@barkhauseninstitut.org
*ahmad.nimr@ifn.et.tu-dresden.de, gerhard.fettweis@tu-dresden.de

***Abstract*—Recent researches indicate that generalized frequency division multiplexing (GFDM) can be regarded as a physical layer waveform core. With proper configuration, it can be used to generate different type of the state of the art waveforms. The GFDM modulation can be realized by means of several discrete Fourier transform (DFT) transforms, a complex multiplier and several memory blocks. However, the conventional implementations of GFDM were limited by the consideration of real-valued symmetric prototype pulse shapes, where even numbers of subcarriers and subsymbols produce singular modulation matrix. The new design of GFDM prototype pulse shape allows all combinations of the parameters, which facilitates an efficient radix-2 implementation. In this paper, we realize a new field programmable gate array (FPGA) implementation of GFDM considering radix-2 parameters. We show that our modem significantly reduces the hardware resource consumption in comparison with conventional implementations. Moreover, it provides high throughput, maintains low-latency, and achieves high accuracy with small fixed-point bit width. Beside that, this implementation is unified for both the time and frequency domains. Furthermore, the hardware design provides additional degrees of freedom with run-time reconfiguration features.**

***Index Terms*—GFDM, hardware implementation, FFT, FPGA, measurement, fixed-point arithmetic.**

I. Introduction

In generalized frequency division multiplexing (GFDM), the available band is divided into subcarriers and each subcarrier is divided into subsymbols. A data symbol modulates a pulse shape that is generated by means of circular shift in the time and frequency domains of a prototype pulse shape [1]. With the degrees of freedom in selecting the number of subcarriers and subsymbols as well as the pulse shape, GFDM can be regarded as a flexible waveform that can be adapted for different requirements. Conventionally, the prototype pulse shape is designed to be localized in the frequency domain to reduce the inter-carrier-interference (ICI). Initially, the prototype pulse shape is considered real-valued symmetric, which poses limitations on the selected parameters. For instance, both even number of subcarriers and subsymbols are avoided as the modulation matrix becomes singular [2]. Influenced by that, the available hardware implementation focuses on odd number of subsymbols [3], [4]. In [5], pulse shape design is developed for different combinations of the parameters. It is shown that a Hermitian symmetric pulse can be used for even number parameters, which initiates the idea of radix-2 implementation.

In [6], GFDM is extended beyond its conventional purpose to cover additional waveforms, such as orthogonal time frequency space (OTFS). It is shown that GFDM can be described in a unified time and frequency-domain structure that consists of four steps. First, the data matrix is precoded in columns and rows with discrete Fourier transform (DFT) matrices. The pulse shaping is performed in a form of windowing, where the window is derived from the prototype pulse. Afterwards, another DFT transform takes place. Finally, the matrix transpose is vectorized into a vector. As a result, the extended GFDM framework provides additional degrees of freedom by bypassing one or more stages required by the conventional one. Moreover, the way the matrix is unfolded to a vector can be customized. For instance, rearranging the time-domain samples can simply generate OTFS signal. On the other hand, the frequency-domain samples can be seen as precoded data that are allocated to orthogonal frequency division multiplexing (OFDM) subcarriers. Putting all together, the extended GFDM provides an efficient tool for waveform design.

The aforementioned features motivate a hardware realization of the extended GFDM framework. In this work, we develop our modem based on the DFT architecture proposed in [7]. Compared with the convolutional-based implementation in [4], our modem benefits from serially pipelined structure, which provides more flexibility for run-time configuration. The functionality is not limited to the conventional GFDM time-domain processing, but the modem can be adjusted during the run-time to perform pure OFDM or DFT-spread-OFDM efficiently. Moreover, the switching between the time-domain and frequency-domain modes is straight forward. We measure the performance in terms of resource usage, latency and fixed-point accuracy. Although the latency is slightly higher than the previous implementation in some cases, the resource usage is significantly decreased. Moreover, our design ensures that all the components are pipelined to achieve maximum throughput. Furthermore, the developed core can run at high clock frequency, which helps in reducing the latency.

The remainder of the paper is organized as follows: Section II provides a short overview of GFDM and its DFT structure,

The work presented in this paper has been performed in the framework of the ORCA project [https://www.orca-project.eu/]. This research was co-financed by public funding of the state of Saxony/Germany.

978-1-7281-3627-1/19/$31.00 © 2019 IEEE

This document is a preprint of: Z. Li, A. Nimr and G. Fettweis, "Implementation and Performance Measurement of Flexible Radix-2 GFDM Modem," in Proceedings of IEEE 5G World Forum (WF-5G 2019), Dresden, Germany, (pp. 130-134), Sep 2019. DOI:10.1109/5GWF.2019.8911718

which is exploited in the implementation. Section III presents the field programmable gate array (FPGA) implementation details. In section IV, we provide performance measurements regarding the hardware usage, latency, and the normalized mean squared error (NMSE). Finally, the paper is concluded in section V.

II. GFDM OVERVIEW

GFDM is a flexible block-based multicarrier modulation scheme. It provides the user with a time-frequency resource block, which has a time duration T and bandwidth B for transmitting a data block with N samples. The available frequency resource consists of K equally spaced subcarriers with the subcarrier spacing $\Delta f = \frac{B}{K}$. On the other hand, the time resource is divided into M equal length segments, which are called the subsymbols. Thus, the subsymbol spacing is $T_{\text{sub}} = \frac{T}{M}$. For a given $N = KM$, the numerology of a user is easily changed with the configuration of the parameters K and M. The conventional GFDM modulation in the time domain is given by

$$x[n] = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} d_{k,m} \mathrm{g}\left[< n - mK >_N\right] \mathrm{e}^{-\mathrm{j}\frac{2\pi kn}{K}}, \tag{1}$$

where $n = 0, 1, \cdots, N-1$ and $\mathrm{g}\left[< n - mK >_N\right] \mathrm{e}^{-\mathrm{j}\frac{2\pi kn}{K}}$ is the time-frequency shifted version of the prototype pulse shape filter $\mathrm{g}[n]$ in the time domain, and $d_{k,m}$ is the data symbol transmitted on the k-th subcarrier and m-th subsymbol. Moreover, the modulation in the frequency domain is given by

$$x[n] = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} d_{k,m} \tilde{\mathrm{g}}\left[< n - kM >_N\right] \mathrm{e}^{-\mathrm{j}\frac{2\pi mn}{M}}, \tag{2}$$

where $\tilde{\mathrm{g}}[n]$ is the prototype pulse represented in the frequency domain [1].

A. Matrix representation

In the conventional matrix representation, the GFDM block is presented by a vector $\boldsymbol{x} \in \mathbb{C}^{N\times 1}$ and its frequency-domain $\tilde{\boldsymbol{x}} = \boldsymbol{F}_N \boldsymbol{x}$, where $\boldsymbol{F}_N$ is the N-DFT matrix. By rearranging the elements of $\boldsymbol{x}$ in a matrix $\boldsymbol{V}_{K,M}^{(\tilde{\boldsymbol{x}})}$ of size $K \times M$, where $\boldsymbol{x} = \mathrm{vec}\left\{\left(\boldsymbol{V}_{M,K}^{(\boldsymbol{x})}\right)^{\mathrm{T}}\right\}$ as in [6], we get the relation

$$\boldsymbol{V}_{M,K}^{(\boldsymbol{x})} = \frac{1}{M} \boldsymbol{F}_M^{\mathrm{H}} \left(\boldsymbol{Z}_{M,K}^{(\boldsymbol{g})} \odot \left[\boldsymbol{F}_M \boldsymbol{D}^{\mathrm{T}} \boldsymbol{F}_K^{\mathrm{H}}\right]\right), \tag{3}$$

where $[\boldsymbol{D}]_{(k,m)} = d_{k,m} \in \mathbb{C}^{K\times M}$ is the data matrix, and $\boldsymbol{Z}_{M,K}^{(\boldsymbol{g})} = \boldsymbol{F}_M \boldsymbol{V}_{M,K}^{(\boldsymbol{g})}$ is denoted as the modulator window. The frequency-domain block can be derived by exchanging the parameters K, M and the DFT direction. Therefore, we get

$$\boldsymbol{V}_{K,M}^{(\tilde{\boldsymbol{x}})} = \frac{1}{K} \boldsymbol{F}_K \left(\bar{\boldsymbol{Z}}_{K,M}^{(\tilde{\boldsymbol{g}})} \odot \left[\boldsymbol{F}_K^{\mathrm{H}} \boldsymbol{D} \boldsymbol{F}_M\right]\right), \tag{4}$$

where $\bar{\boldsymbol{Z}}_{M,K}^{(\boldsymbol{g})} = \boldsymbol{F}_K^H \boldsymbol{V}_{K,M}^{(\tilde{\boldsymbol{g}})}$, and $\tilde{\boldsymbol{x}} = \mathrm{vec}\left\{\left(\boldsymbol{V}_{K,M}^{(\tilde{\boldsymbol{x}})}\right)^{\mathrm{T}}\right\}$. Accordingly, the GFDM modulator has a unified time and frequency architecture, as illustrated in Fig. 1. In this diagram, the matrix $\boldsymbol{W}_{\text{tx}}$ refers to the window either in the time or frequency domain. The implementation of the modem mainly requires three DFT/inverse DFT (IDFT) blocks and one complex multiplication. An additional DFT block is needed for the frequency-domain processing. The demodulator is realized by reversing the order of the blocks.

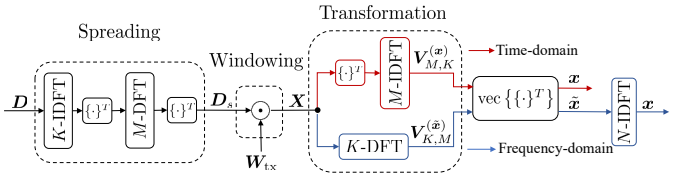


Fig. 1: Unified architecture of GFDM modulator.

III. RADIX-2 GFDM MODEM IMPLEMENTATION

In this section, we present the FPGA implementation of the architecture shown in Fig. 1 using LabVIEW Communications System Design Suite. Fig. 2 presents the more detailed block diagram of the modulator and demodulator considering radix-2 values of K, M parameters. It is worth mentioning that the component modules of the modulator and demodulator are the same. These are flexible fast Fourier transform (FFT) module, scaler module, matrix transpose module, complex multiplier module, and sample mapping module. The demodulator can be distinguished from the modulator with the order of the component modules. Furthermore, all the modules shown in the block diagram can be switched off independently to provide the extended flexibility of GFDM. In the following parts, we focus on the specifications of each component.

A. Flexible FFT Module

The flexible FFT module performs the N-point DFT or IDFT operation, where N is a radix-2 number ranging from 1 to the GFDM data block size N_{data}. The Xilinx FFT intellectual property (IP) Core is chosen for the DFT operation with size larger than 8. The bit width of the input samples is set to 22 bits to enhance the accuracy of the implementation. The architecture mode is set to pipelined steaming I/O mode to increase the throughput of the implementation. Moreover, the optimization option is selected to target resource minimization. For the 2- and 4-point DFT or IDFT operation, an FFT core is implemented based on the Cooley-Tukey algorithm, which also has a pipelined structure to maximize the throughput and reduce the timing requirement of the flexible FFT module.

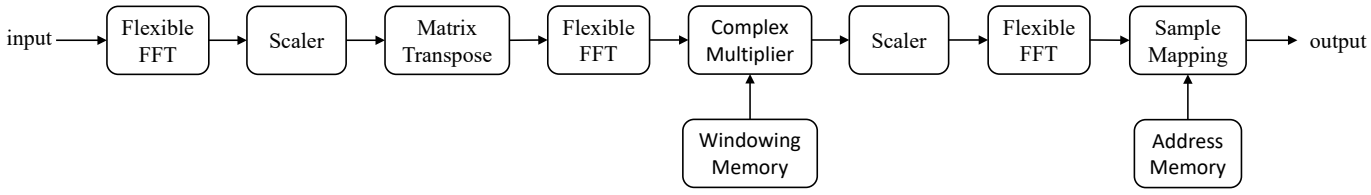
B. Scaler Module

According to the DFT and IDFT calculation

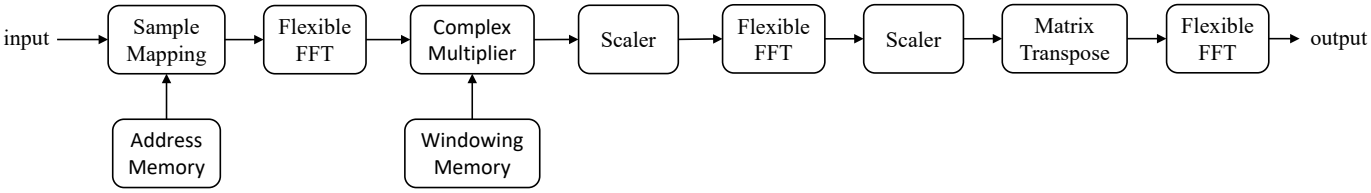
$$X[k] = \sum_{n=0}^{N-1} x[n] \mathrm{e}^{-\mathrm{j}\frac{2\pi kn}{N}}, x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \mathrm{e}^{\mathrm{j}\frac{2\pi nk}{N}},$$

the samples at the output of DFT/IDFT can be up to N times larger than the input samples, which means, in the radix-2 case, $\log_2 N$ bits are additionally required in order to represent data at the output without clipping. However, the bit width of the digital-to-analog converter (DAC) and

This document is a preprint of: Z. Li, A. Nimr and G. Fettweis, "Implementation and Performance Measurement of Flexible Radix-2 GFDM Modem," in Proceedings of IEEE 5G World Forum (WF-5G 2019), Dresden, Germany, (pp. 130-134), Sep 2019. DOI:10.1109/5GWF.2019.8911718



(a) Modulator



(b) Demodulator

Fig. 2: GFDM modem implementation block diagram.

analog-to-digital converter (ADC) before and after the radio-frequency (RF) component is limited, thus, the samples need to be down-scaled after the DFT/IDFT operation to prevent data clipping. Since the flexible FFT module performs radix-2 FFT operation, a bit shifter is implemented for the data scaling to reduce the resource consumption.

C. Matrix Transpose Module

Due to the serial data streaming on the FPGA, two methods are applicable to index the elements in a matrix as visualized in Fig. 3. One is indexing the matrix column-by-column, defined as the vectorization of the matrix, and the other is indexing row-by-row, which is equivalent to vectorizing the transposed matrix. The matrix transpose on the FPGA is realized by altering the way to forward the matrix, i.e. to provide the row-wise upcoming matrix in a column-wise reading or vice versa. As depicted in Fig. 4, the matrix transpose operation is

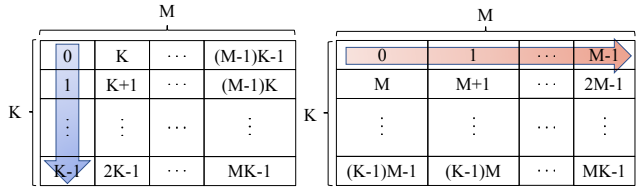


Fig. 3: Matrix indexing column-wise/ row-wise.

achieved by controlling the memory address of the upcoming elements in the memory. Assuming Q is the length of the row (column) of a matrix upcoming row-wisely (column-wisely) to the transpose module, and let P be the length of the column (row), i.e. $P = \frac{N}{Q}$, where N is the block size. The address $\mathrm{addr}[n]$ generated by the address counter unit for the n-th element is given by

$$\mathrm{addr}[n] =< n >_Q P + \mathrm{int}(\frac{n}{Q}) \ , \ n = 0, 1, \cdots, N-1, \quad (5)$$

where $< n >_Q$ is the modulo operator representing the index of the row (column) and $\mathrm{int}(\frac{n}{Q})$ is the integer division, which represents the index of the column (row).

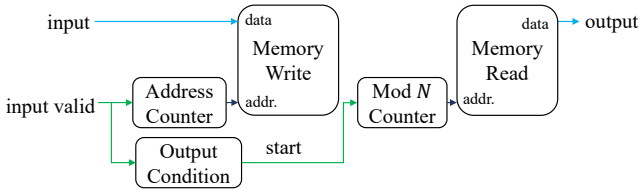


Fig. 4: Block diagram of Matrix Transpose module.

The output condition unit monitors the number of stored samples and triggers the forwarding process with the calculated condition to reduce the latency of the matrix transpose module. As illustrated in Fig. 5, when the $(Q(P-1)+1)$-th element is being stored in the memory, the $(Q-1)$-th elements can be provided to the next stage. Thus, at the moment of storing the $(N-Q-P+3)$-th element, the memory reading process can be triggered. The bit width of the memory is

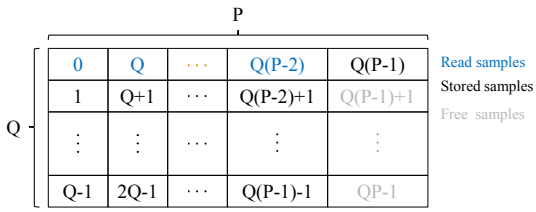


Fig. 5: Visualization of output condition.

configured the same as that of the flexible FFT input. With that, the resource consumption is reduced while maintaining the accuracy. Moreover, a double-page memory is implemented to allow the matrix transpose module to store a new matrix and forward the previous matrix simultaneously, which ensures the maximum throughput.

This document is a preprint of: Z. Li, A. Nimr and G. Fettweis, "Implementation and Performance Measurement of Flexible Radix-2 GFDM Modem," in Proceedings of IEEE 5G World Forum (WF-5G 2019), Dresden, Germany, (pp. 130-134), Sep 2019. DOI:10.1109/5GWF.2019.8911718

D. Windowing Module

The windowing module is implemented with Xilinx Complex Multiplier IP Core, whose fully-pipelined architecture maintains the throughput. The bit width of the data is optimized, targeting the balance of resource consumption and processing accuracy. The modulation and demodulation windows for different configurations can be prestored in a memory to reduce the configuration time.

E. Sample Mapping Module

This module is used to map the generated samples to the final vector. This implies permutation for the time-domain samples, whereas it is equivalent to subcarrier allocation for the frequency-domain samples. The structure of the sample mapping module is based on that of the matrix transpose module. However, the addresses in this case are prestored and can be reconfigured. To facilitate pilot multiplexing, a header field is added to the address to tell whether the current sample needs to be read from the pilot source or from the modem output. Moreover, the sample mapping module supports the mapping procedure for the multi-user case, where a small size GFDM modulation in the frequency domain can be performed, and the blocks of different users are mapped into the final vector as represented in Fig. 6.

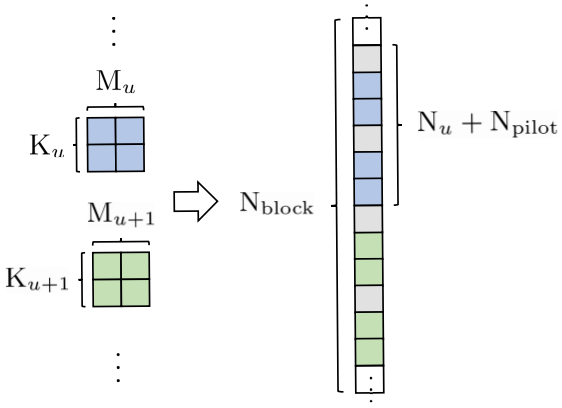


Fig. 6: Mapping for multi-user case.

IV. Evaluation

In this section, the evaluation of the implementation is presented, including the hardware usage on the FPGA, the latency, and the NMSE compared to floating point arithmetic. All the tests are performed on NI USRP-2954. Moreover, the maximum block size $N = KM$ of GFDM is set to 2048.

A. Hardware usage

The hardware utilization is evaluated by the LabVIEW Communications Compile Worker, which uses the Xilinx Vivado compiler, including the number of registers, digital signal processor (DSP) units, block RAM, and lookup tables (LUTs). The hardware usage of modulator alone is presented in Fig. 7, whereas Fig. 8 presents the total hardware usage of the modulator and demodulator.

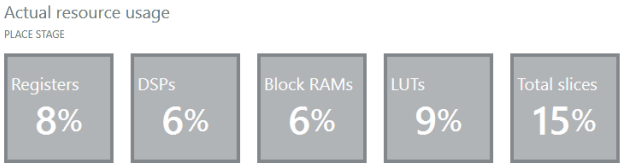


Fig. 7: Hardware usage the modulator.

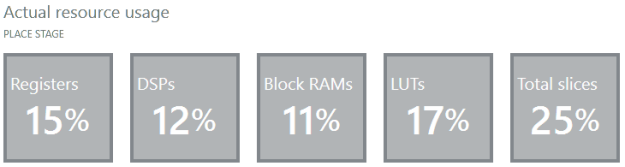


Fig. 8: Hardware usage of the complete modem.

The flexibility of waveform generation comes with a price, the proposed implementation consumes more registers compared with the OFDM case, where only two FFT modules are responsible for the modulation and demodulation. Also, the DSP consumption is higher, since it has three FFT stages, and each stage is capable of performing 2048-point FFT to maximize the flexibility of the implementation.

B. Latency

In this paper, the latency is defined as

$$T_{\text{latency}} = T_{\text{loading}} + T_{\text{processing}} + T_{\text{unloading}} \ [\text{clock cycles}], \quad (6)$$

where T_{loading} and $T_{\text{unloading}}$ is the loading and unloading time of a data block into or out of the pipelined module. $T_{\text{processing}}$ is the actual computation time. The theoretical latency of the implementation performing GFDM time-domain modulation is given by

$$T_{\text{TD}} = 3N + T_{K\text{-FFT}} + 2T_{M\text{-FFT}} + 2T_{\text{s}} + T_{\text{c}} + M + 3, \quad (7)$$

whereas the latency of the frequency-domain modulation is given by

$$T_{\text{FD}} = 3N + T_{M\text{-FFT}} + 2T_{K\text{-FFT}} + 2T_{\text{s}} + T_{\text{c}} + K + 3, \quad (8)$$

where the $T_{N\text{-FFT}}$, T_{s}, and T_{c} are the processing latency of the N-point FFT, the scaler, and the complex multiplier. With the well-pipelined structure, the implementation can be compiled with up to 250 MHz clock frequency on Xilinx Kintex-7 FPGA in the USRP. The result of the latency is given in microsecond using 250 MHz clock on the FPGA. Fig. 9 presents the latency of the GFDM time-domain modulator alone and the total latency of GFDM time-domain modulator and demodulator. The latency of generating the frequency-domain samples can be also derived from Fig. 9, since it only switches the role of K and M.

This document is a preprint of: Z. Li, A. Nimr and G. Fettweis, "Implementation and Performance Measurement of Flexible Radix-2 GFDM Modem," in Proceedings of IEEE 5G World Forum (WF-5G 2019), Dresden, Germany, (pp. 130-134), Sep 2019. DOI:10.1109/5GWF.2019.8911718

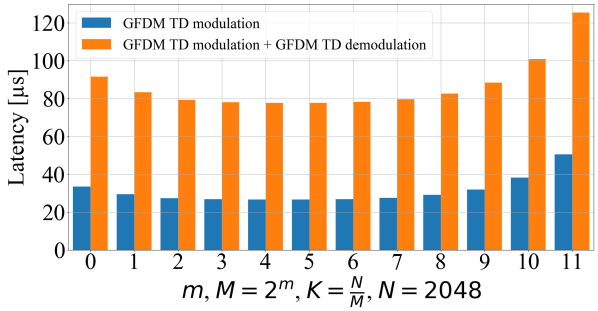


Fig. 9: Latency measurement.

C. NMSE

The NMSE represents the error produced by the fix-point arithmetic on the FPGA compared with the floating-point arithmetic on the host computer. The NMSE is given by

$$\text{NMSE} = \frac{\text{E}\left[|x_{\text{ref}} - x_{\text{FPGA}}|^2\right]}{\text{E}\left[|x_{\text{ref}}|^2\right]}, \tag{9}$$

where x_{ref} is the reference result provided by host PC performing floating-point computation, and x_{FPGA} is the result from FPGA using fix-point. The most critical configuration is chosen for the NMSE measurement, i.e. all the modules are switched on and the modulation matrix is set to be a random matrix. The measured results of the GFDM time-domain modulator are shown in Fig. 10. The error caused by the fix-

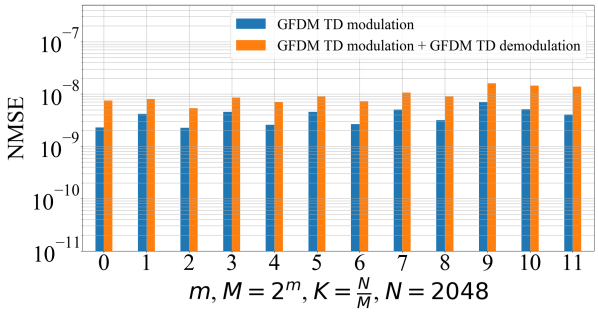


Fig. 10: NMSE of FPGA fixed-point vs floating-point.

point arithmetic can be interpreted as additive noise. The signal to noise ratio (SNR) is given by the SNR $= 1/\text{NMSE}$. Considering a system using the quadrature amplitude modulation (QAM) symbols, the bit error rate (BER) assuming additive white Gaussian noise (AWGN) is estimated by

$$P_{b,S} = \frac{2\left(\sqrt{S}-1\right)}{\sqrt{S}\text{ld}\left(S\right)}\text{erfc}\left(\sqrt{\frac{3}{2\left(S-1\right)}10^{0.1SNR}}\right), \tag{10}$$

where S is the order of the constellation. Form Fig. 10, it can be seen that the worst BER introduced by the modulation and demodulation using 256-QAM is negligible.

V. Conclusion

Our proposed radix-2 implementation is based on the DFT based architecture of GFDM. Comparing with the current designs, it abandons the parallel convolution modules and converts them into a serial pipelining structure, which reduces the hardware consumption for the modulation with a larger M in the time domain. The implementation is accomplished with LabVIEW communications design suite as an FPGA design. This implementation supports run-time configuration. It can be configured as GFDM modulator in the time or frequency domain during the run-time. Moreover, the components in the implementation can be enabled or disabled independently to ensure flexibility. With the extended flexibility, it can provide the processing of specific configurations such as OFDM and singe-carrier FDMA (SC-FDMA). The implementation has a pipelining structure and works with 250 MHz clock frequency in the USRP, which maximizes the throughput and reduces the latency.

Also, we provide the evaluation of the implementation, including the hardware usage, the latency, and the NMSE. Although the implementation consumes more hardware resource compared with the OFDM system, it can be treated as a physical layer waveform core and provides more flexibility in run-time. The maximum latency of the implementation including GFDM time-domain modulation and demodulation is around 125μs when the GFDM block size is 2048. Furthermore, the comparison with the floating-point arithmetic shows that the error caused by the implementation is negligible using the 256-QAM mapping scheme, which is the highest constellation order of LTE system.

References

[1] N. Michailow *et al.*, "Generalized frequency division multiplexing for 5th generation cellular networks," *IEEE Trans. Commun.*, vol. 62, no. 9, pp. 3045–3061, Sep 2014.

[2] M. Matthé *et al.*, "Generalized frequency division multiplexing in a gabor transform setting," *IEEE Commun. Lett.*, vol. 18, no. 8, pp. 1379–1382, 2014.

[3] M. Danneberg *et al.*, "Flexible GFDM Implementation in FPGA with Support to Run-Time Reconfiguration," in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, Sept 2015, pp. 1–2.

[4] TU-Dresden, "Flexible transceiver implementation." [Online]. Available: http://owl.ifn.et.tu-dresden.de/GFDM/

[5] A. Nimr *et al.*, "Optimal Radix-2 FFT Compatible Filters for GFDM," *IEEE Communications Letters*, vol. 21, no. 7, pp. 1497–1500, 2017.

[6] A. Nimr *et al.*, "Extended GFDM Framework: OTFS and GFDM Comparison," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec 2018, pp. 1–6.

[7] A. Nimr, M. Chafii, and G. P. Fettweis, "Unified low complexity radix-2 architectures for time and frequency-domain GFDM modem," *IEEE Circuits and Systems Magazine*, vol. 18, no. 4, pp. 18–31, Fourthquarter 2018.