Bit Precision Study of A Non-Orthogonal Iterative Detector with FPGA Modelling Verification

Tongyang Xu and Izzat Darwazeh

Department of Electronic and Electrical Engineering, University College London, London, UK
Email: t.xu@ee.ucl.ac.uk, i.darwazeh@ucl.ac.uk

***Abstract*—Much work has been done on a non-orthogonal signal termed spectrally efficient frequency division multiplexing (SEFDM). Due to its self-created inter carrier interference (ICI), signal detection is complicated. A linear detector named iterative detection (ID) detector shows better bit error rate (BER) performance and complexity trade-off than other linear detectors. Therefore, this work shows the first time hardware modelling of the ID detector at the register transfer level (RTL) stage. The impact of bit precision on the system performance is studied at the beginning. Then, an RTL model is designed with results showing competitive fixed-point performance which are comparable to Matlab floating-point results. Verification work is operated in a co-simulation environment through comparison between fixed-point Matlab results and ISim (a hardware modelling software from Xilinx Inc.) simulation results. Their results are consistent indicating the hardware model is correct.**

***Index Terms*—Multicarrier communications, spectral efficiency, OFDM, SEFDM, 5G, non-orthogonal, RTL, hardware.**

I. Introduction

In future 5^{th} generation (5G) networks [1][2], non-orthogonal signals will become a dominate technical direction with different applications gradually appearing. One promising 5G waveform, termed spectrally efficient frequency division multiplexing (SEFDM) [3][4], can improve spectral efficiency by compressing signal bandwidth compared to orthogonal frequency division multiplexing (OFDM). The generation of SEFDM signals has been fully studied and implemented in hardware such as [5][6]. However, signal detection at the receiver is still challenging both in software modelling and hardware implementation. Maximum likelihood (ML) detector shows optimal performance but with high complexity. Its complexity reduced algorithm, termed sphere decoding (SD) [7], shows attractive performance with reasonable complexity. It has been accepted in multiple input multiple output (MIMO) systems with very large scale integration (VLSI) hardware implementation in [8]. However, the random complexity limits its widely use. Its fixed complexity version, termed fixed sphere decoding (FSD) [9], shows competitive performance while it is still limited to small size MIMO systems. Therefore, an efficient linear detector, which doesn't require Euclidean norm operations, seems to be a practical solution. The hardware design of the linear truncated singular value decomposition (TSVD) detector has been reported in [10] due to its low complexity and improved performance compared to zero forcing (ZF) detector. Moreover, theoretical work in [11] proves that the iterative detection (ID) outperforms the TSVD significantly. However, its complexity is analyzed based on counting the number of basic arithmetic operations which gives a poor estimate of hardware resource utilizations. The only way to get an accurate complexity estimate is to actually implement a system. But realizing a system in hardware is time consuming and sometimes not necessary. Therefore, this work shows, for the first time, a field programmable gate array (FPGA) RTL modelling of the ID detector.

II. System Model and Iterative Detection

The discrete format of SEFDM signals is expressed as

$$X[k] = \frac{1}{\sqrt{Q}} \sum_{n=0}^{N-1} s_n \exp[\frac{j2\pi nk\alpha}{Q}] \qquad (1)$$

where α is the bandwidth compression factor defined as $\alpha = \Delta f T$, where Δf denotes the frequency distance between adjacent sub-carriers, T is the period of one SEFDM symbol. $X[k]$ indicates time samples with index $k = [0, 1, ..., Q-1]$, $Q = \rho N$ is the number of time samples with oversampling factor $\rho \geq 1$, s_n is input symbol vector with $n \in [0, ..., N-1]$ and $\frac{1}{\sqrt{Q}}$ is a normalization factor. Its matrix format is

$$X = \mathbf{F}S \qquad (2)$$

where $\mathbf{F}$ is a $Q \times N$ sub-carrier matrix with elements equal to the SEFDM complex sub-carriers $e^{\frac{j2\pi nk\alpha}{Q}}$.

At the receiver after additive white Gaussian noise (AWGN), the signal model in matrix format is

$$R = \mathbf{C}S + Z \qquad (3)$$

where R is an N-dimensional vector of demodulated symbols, $\mathbf{C}$ is an $N \times N$ correlation matrix defined by the multiplication of $\mathbf{F}$ and its conjugate $\mathbf{F}^*$ and Z is an N-dimensional vector of AWGN noise samples correlated with $\mathbf{F}^*$.

The principle of iterative detection has been studied in [11]. The basic idea is to cancel inter carrier interference (ICI) iteratively using a gradually reduced decision interval, $\Delta A = 1 - m/v$, where m is the m^{th} iteration and v is the total iteration number. The interval is reduced gradually until zero in the iteration process and hard decision is made at the end. The iterative operation is expressed as

$$S_n = R - (\mathbf{C} - \mathbf{e})S_{n-1}, \tag{4}$$

where $\mathbf{e}$ is an $N \times N$ identity matrix, S_n is an N-dimensional vector of recovered symbols after n iterations and S_{n-1} is an N-dimensional vector of estimated symbols after $n-1$ iterations.

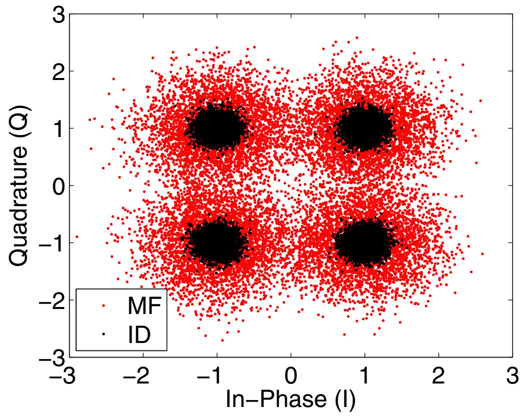


Figure 1. Matched filtering (red) and iterative detected (black) constellation diagrams for 4QAM-SEFDM with α=0.8 at E_b/N_o=14dB.

More details of the iteration operation are referred to [11]. The benefits of using the iterative detection scheme are demonstrated via constellation points in Fig. 1. The red dots indicate matched filter (MF) results, which are interfered significantly. The black dots are much better due to the iterative removal of self-created ICI using the iterative detection scheme.

III. Analysis on the Fixed-Point Precision

Prior to the hardware modelling, Matlab simulation is required to get an initial result for the purpose of comparison. In a real hardware system, the cost of floating-point arithmetic is significant since it consumes more hardware resources and power than a fixed-point operation. Therefore, a fixed-point representation is favoured in hardware design. In terms of the Matlab simulation, the fixed-point representation is used since it is very close with an actual hardware result. In order to get the result, it is necessary to understand the composition of a fixed-point symbol. A fixed-point symbol is composed of three parts: the first part determines the sign (positive or negative); the second part is its integer part while the last stands for the decimal part. In this section, a concept termed 'precision' is defined as the number of bits represented in the decimal part.

The precision plays an important role in the performance and complexity of a system. Generally speaking, the higher precision, the better performance but at the expense of higher complexity. Therefore, this section firstly investigates the effects of various precisions and illustrates their performance through Matlab simulations. The error performance is tested by evaluating 10,000 4QAM-SEFDM symbols through an AWGN channel.

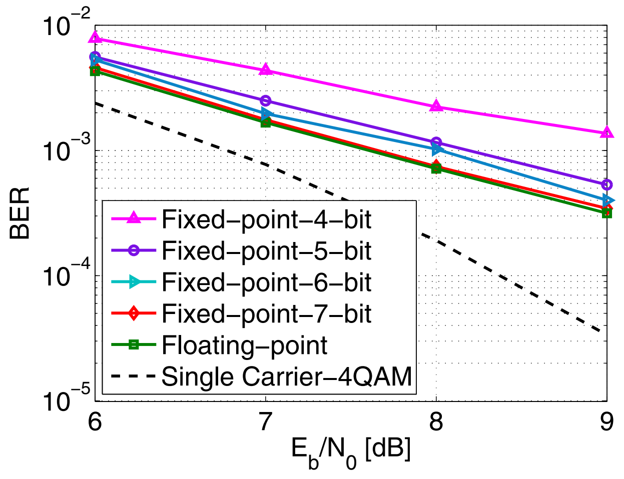


Figure 2. Fixed-point and floating-point performance of ID detector with 4QAM-SEFDM symbols at α=0.8 for various bit precisions.

Fig. 2 presents both fixed-point and floating-point results modulated with 4QAM symbols at precisions ranging from 4 bits to 7 bits. It is evident that 7 bits are sufficient for a fixed-point representation to approach the floating-point result. It should be noted that there is a 0.5 dB degradation when the precision goes down to 5 bits. However, the 0.5 dB gap is negligible considering that a lower precision can save a lot of hardware resources. Therefore, in the following hardware modelling, 5 bits are used to be the default precision for a fixed-point representation.

IV. Design Architecture

The hardware modelling in this section is based on the first principle that a parallel and pipelined structure is adopted. This kind of architecture has a maximum throughput while the complexity level is relatively high.

The conversion from a floating-point representation to a fixed-point one is determined by the level of accuracy and the dynamic range [10]. The dynamic range is defined as the ratio of the maximum symbol amplitude to the minimum one. The accuracy of a symbol is decided by its decimal part. Therefore, the conversion work involves scaling the decimal part by multiplying with a scaling factor [10]. In practice, the scaling factor is limited by the resolution of a hardware component. In our design, it is determined by the available resources in the FPGA chip. Assuming sufficient hardware resources are available and the effects of different scaling factors have been emulated and illustrated in Fig. 2 where the 5-bit decimal representation is proved to be a compromising precision.

Noticing that there is no need to implement a product using a multiplier. This operation can be replaced by a simple shift and add operation. This approach brings resource/power savings and improved processing speed. Therefore, the scaling factor here is defined as 2^γ where γ is the number of bits standing for the resolution of a decimal part. For a binary representation, being multiplied by a scaling factor 2^γ indicates shifting all the binary bits left by γ bit positions with zeros inserted in the old positions. This shift operation is much simpler than a high precision multiplier, meanwhile, it is cost efficient and power efficient.

In this section, the length of one fixed-point symbol is set to be 8 bits where the first bit determines the sign (positive or negative) of a symbol; the next 2 bits represent the integer part and the last 5 bits (precision γ=5) stand for the decimal part. The software used here is Xilinx ISE and its programming language is Verilog HDL. Since the purpose of this hardware modelling is to test the functionality of the ID detector, therefore, system optimization is not considered. Throughout the modelling, a parallel design principle is followed leading to a high throughput at the cost of high resource utilizations.

The use of the first bit for the symbol sign is sufficient since only two options exist. A positive symbol indicates a bit '0' while a negative symbol is represented by a bit '1'. However, in terms of the integer part, two bits are used with an amplitude ranging from '0' to '3'. If the integer part of a number exceeds the defined maximum absolute amplitude '3', an overflow occurs leading to unexpected errors. The absolute amplitude of an ideal 4QAM symbol is '1', with small fluctuations from noise or interference, its absolute amplitudes maybe still within the range. Therefore, the use of two bits for the integer part has little effect on 4QAM symbols. The rest of bits are reserved for the representation of a decimal value ranging from '0' to '1'. The effect of decimal precision has been studied in the preceding section.

Fig. 3 shows the RTL architecture of the ID detector. Seven modules including $\mathbf{MUX\_2\_TO\_1}$, $\mathbf{ID\_FSM}$, $\mathbf{INITIAL\_FSM}$, $\mathbf{CORE}$, $\mathbf{HOLD}$, $\mathbf{IUPUT\_PORT}$ and $\mathbf{OUTPUT\_PORT}$ are illustrated. The functions of each module are described in the following.

- **MUX_2_TO_1.**
  This component is responsible for controlling the source of incoming symbols. One source comes from the $\mathbf{IUPUT\_PORT}$ while the other one is the feedback estimates from the $\mathbf{CORE}$. The signal detection in the rest of modules results in processing delays, therefore, $\mathbf{MUX\_2\_TO\_1}$ has to block the incoming symbols from the $\mathbf{IUPUT\_PORT}$ before the previous symbol is completely detected. A control signal termed 'SEL' coming from the $\mathbf{ID\_FSM}$ is used to switch the incoming symbols between the two modules.
- **ID_FSM.**
  This small component has two functions based on the output signal 'SEL'. The first function is to control the switching operation in the $\mathbf{MUX\_2\_TO\_1}$. The second one is to decide the symbol output time of the module $\mathbf{HOLD}$ where its output time is consistent with the symbol input time of the $\mathbf{IUPUT\_PORT}$.
- **INITIAL_FSM.**
  This module provides a system parameter termed uncertainty interval ΔA [11] for the $\mathbf{CORE}$. The uncertainty interval ΔA is reduced iteratively until zero with the assistance of a counter inside. More information about the function of this parameter is referred to [11].
- **CORE.**
  This module is the key component that performs the iterative detection where a large amount of computations are required here. The uncertainty interval ΔA is required to determine a decision boundary. After the first iteration, the estimated symbols derived from the $\mathbf{CORE}$ are fed back to the $\mathbf{MUX\_2\_TO\_1}$ waiting for the next operation. After receiving the control signal 'SEL' and the updated uncertainty interval ΔA, the estimates stored inside the $\mathbf{MUX\_2\_TO\_1}$ are transferred back to the $\mathbf{CORE}$ for the second iteration. Therefore, from the first iteration to the last but one, based on the control signal 'SEL' and the uncertainty interval ΔA, the estimates are fed back and fed forward between $\mathbf{CORE}$ and $\mathbf{MUX\_2\_TO\_1}$. At the last iteration, the estimates are directly transferred to the $\mathbf{HOLD}$ module and finally output.
- **HOLD.**
  This module aims to hold or output detected symbols based on the control signal 'SEL'. Prior to the completion (i.e. the last iteration) of an signal detection, output symbols are blocked from this module. At the last iteration, according to the signal 'SEL', the symbols delivered from the $\mathbf{CORE}$ are output through this component.

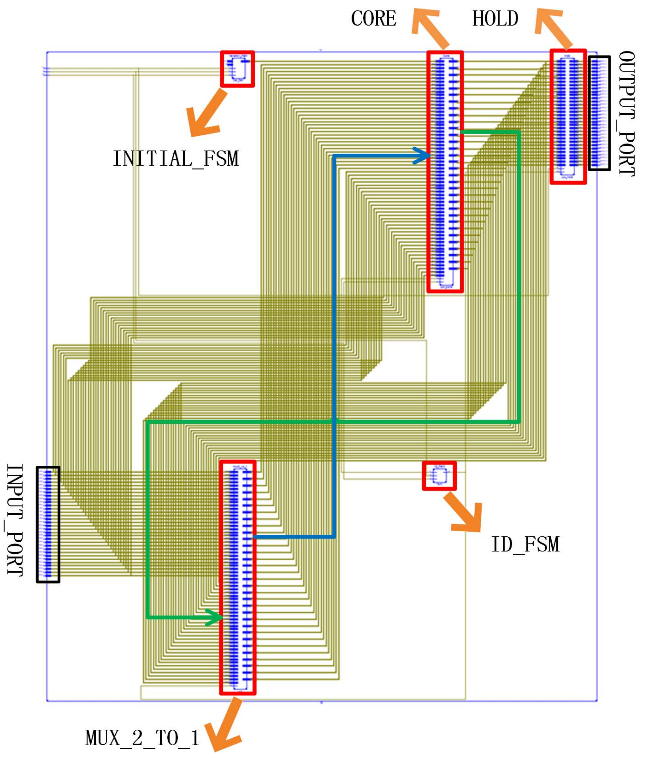


Figure 3. RTL architecture of the ID detector.

Table I
TIMING PERFORMANCE OF THE ID DETECTOR

Timing Performance of the ID Detector	
One iteration period	5 clock cycles
Number of clock cycles per SEFDM symbol	32×5=160 clock cycles
Maximum frequency	168.223 MHz
Minimum period of one clock cycle	5.944 ns
Bits per SEFDM symbol	16×2=32 bits
Detection period for one SEFDM symbol	5.944×160=959.04 ns
Throughput	32/959.04=33.37 Mbit/s

Table II
SYNTHESIS RESOURCE UTILIZATION OF THE ID DETECTOR

Resource Utilization Summary (Synthesis)			
Logic	Used	Available	Utilization
Slice registers	3969	301,440	1%
Slice LUTs	16,147	150,720	10%
DSP48E1s	768	768	100%

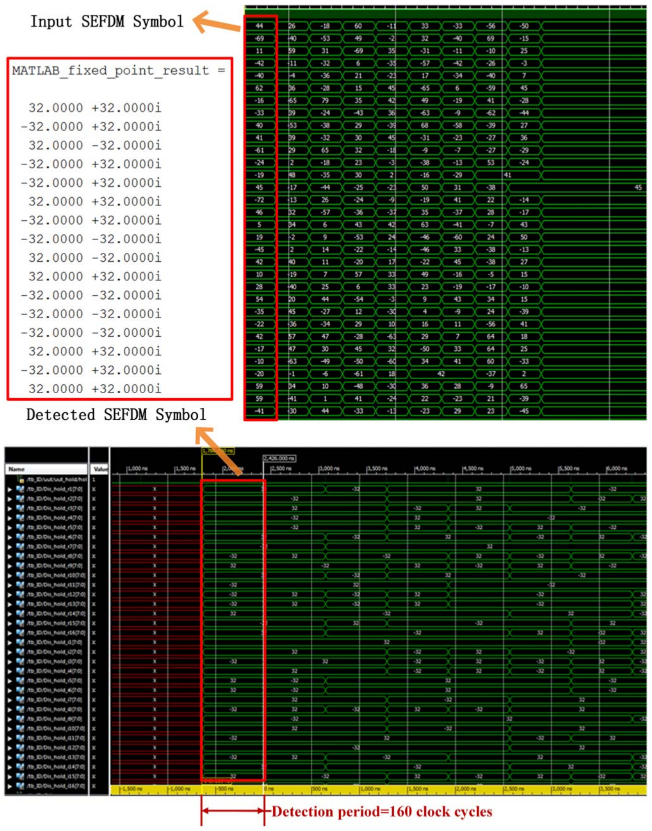
V. PERFORMANCE AND COMPLEXITY ASSESSMENTS

This is the first attempt to model the ID detector in an FPGA RTL scenario. Therefore, at the first stage, a simple system is configured. Starting with a functionality evaluation, the number of sub-carriers is set to be 16; bandwidth compression factor is α=0.8 and modulation scheme is 4QAM. The optimal bit precision is considered to be 5 bits, which is trade-off between performance and complexity. Multiplication can be replaced by using a shift and add operation for the purpose of resource/power saving. Thus, iteration number in this case is set to be $2^5 = 32$, which is larger than a normal one (i.e. 20). Hence, the performance can be guaranteed.

In the design of practical systems, an M-dimensional complex computation can be simplified through the real valued decomposition (RVD) where an M-dimensional complex matrix/vector can be decomposed into an equivalent 2M-dimensional real matrix/vector. Therefore, (3) can be converted to

$$\begin{pmatrix}\Re(R)\\ \Im(R)\end{pmatrix} = \begin{pmatrix}\Re(\mathbf{C}) & -\Im(\mathbf{C})\\ \Im(\mathbf{C}) & \Re(\mathbf{C})\end{pmatrix}\begin{pmatrix}\Re(S)\\ \Im(S)\end{pmatrix} + \begin{pmatrix}\Re(Z)\\ \Im(Z)\end{pmatrix} \quad (5)$$

The timing performance and resource utilization of the ID detector in a hardware modelling environment are presented in Table. I and Table. II, respectively. It is noted that the resource utilization of DSP48E1 is 100%. DSP48E1 is a hard core in the FPGA chip. This element can realize multiplications between two real or complex numbers in a more optimized way. The function of this element can be replaced using slice LUTs, which will save a lot of hard cores in the FPGA chip. One concern about the ID hardware modelling is its iterative architecture. Employing parallel processing architecture consumes a large amount of hardware resources. Therefore, reusing hardware resources is quite important. The second concern is the processing delay. According to the aforementioned modelling, one iteration requires 5 clock cycles. This latency leads to a pipelined delay for the ID detector. With the increase of iteration numbers, the processing delay becomes more significant. Thus, reducing processing time for each symbol or processing more symbols simultaneously would be crucial for a practical ID detector.



```
MATLAB_fixed_point_result =

  32.0000 +32.0000i
 -32.0000 +32.0000i
  32.0000 -32.0000i
 -32.0000 +32.0000i
 -32.0000 +32.0000i
  32.0000 +32.0000i
 -32.0000 +32.0000i
 -32.0000 -32.0000i
  32.0000 -32.0000i
  32.0000 +32.0000i
 -32.0000 -32.0000i
 -32.0000 -32.0000i
 -32.0000 -32.0000i
  32.0000 +32.0000i
 -32.0000 +32.0000i
  32.0000 +32.0000i
```

Figure 4. Illustration of input and detected SEFDM fixed-point symbols of the ID detector from Xilinx Isim waveform generator.

Hardware RTL modelling results are shown in Fig. 4 where the screenshots of input symbols and detected symbols are illustrated together with the theoretical result generated from Matlab fixed-point simulations. In the first inset (right top) of Fig. 4, due to the use of 4QAM modulation scheme, one complex SEFDM symbol (16 aligned 4QAM symbols) can be decomposed into 32 real symbols using the RVD. The first 16 symbols represent real parts while the rest of symbols stand for imaginary parts. Each original floating-point symbol has been converted to its corresponding fixed-point symbol multiplying with $2^5 = 32$ (5 bits for the decimal resolution). In other words, the integers illustrated in the first inset are the symbols after shifting left by 5 bit positions with zeros inserted in the old positions. After a predefined iteration number, the detected symbols are illustrated in the second inset (bottom). The value of '-32' indicates '-1' and '32' represents '1'. The third inset (left top) presents one detected SEFDM symbol through Matlab fixed-point simulation which emulates an equivalent hardware modelling. Through a careful comparison, it is evident that the two results are identical indicating a successful hardware modelling.

To verify the effectiveness of the RTL modelled iterative

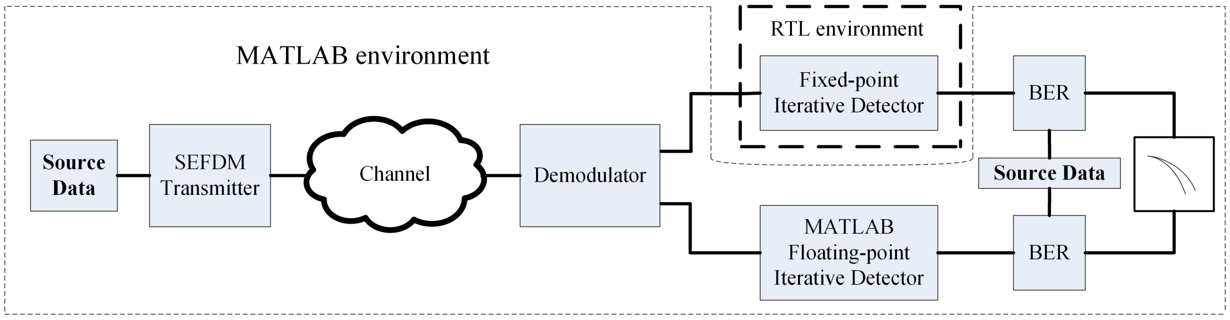


Figure 5. Verification methodology of the fixed-point iterative detector.

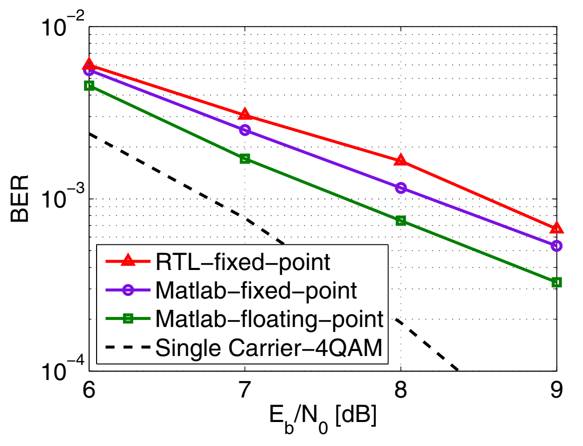


Figure 6. Comparison of RTL result using 5-bit precision and Matlab floating-point and fixed-point results.

detector, a co-simulation methodology is employed as shown in Fig. 5. Source data is generated, modulated, AWGN distorted and demodulated in the Matlab environment. Two copies of the demodulated symbols are fed to the RTL environment and Matlab environment, respectively. Xilinx ISE simulator within the RTL environment generates fixed-point results while Matlab provides floating-point results. BER is calculated for each result and two BER curves are overlapped in the same figure to evaluate their similarities.

Fig. 6 shows the overlapped BER performance of RTL fixed-point result and Matlab emulated fixed/floating-point results. As mentioned before, considering the trade-off between performance and complexity, 5-bit precision is employed. Thus, a performance gap, approximately 0.5dB at E_b/N_o=1e-3, exists between the theoretical floating-point result and its fixed-point one. However, there is a 0.3dB gap between the Matlab fixed-point result and RTL fixed-point one at E_b/N_o=1e-3. This is due to the overflow issue mentioned in section III. This challenge exists in the RTL emulation rather than the Matlab simulation.

VI. Conclusions

This paper studied bit precision on the performance of an iterative SEFDM detector. The trade-off between bit precision and system performance has been considered. An optimal design should reduce the complexity by minimizing the bit precision while meeting the BER performance requirement. Due to the attractive performance and relatively low complexity, a hardware model of the iterative detector is investigated here for the purpose of future practical applications. The theoretical throughput of the modelled ID detector can reach 33.37 Mbit/s. Furthermore, the model of the ID detector is verified through a co-simulation methodology. The RTL modelling result is similar to the theoretical fixed-point Matlab simulation results confirming the correctness of the hardware model at a pre-implementation stage.

References

[1] F.-L. Luo and C. Zhang, *Signal Processing for 5G: Algorithms and Implementations*. Wiley, 2016.

[2] W. Xiang, K. Zheng, and X. Shen, *Key Enabling Technologies for 5G Mobile Communications*. Springer, 2016.

[3] M. Rodrigues and I. Darwazeh, “A spectrally efficient frequency division multiplexing based communications system,” in *Proc. 8th Int. OFDM Workshop*, Hamburg, 2003, pp. 48–49.

[4] T. Xu and I. Darwazeh, “Bandwidth compressed carrier aggregation,” in *IEEE ICC 2015 - Workshop on 5G & Beyond - Enabling Technologies and Applications (ICC’15 - Workshops 23)*, London, United Kingdom, Jun. 2015, pp. 1107–1112.

[5] M. Perrett and I. Darwazeh, “Flexible hardware architecture of SEFDM transmitters with real-time non-orthogonal adjustment,” in *Telecommunications (ICT), 2011 18th International Conference on*, May 2011, pp. 369–374.

[6] P. Whatmough, M. Perrett, S. Isam, and I. Darwazeh, “VLSI architecture for a reconfigurable spectrally efficient FDM baseband transmitter,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 59, no. 5, pp. 1107–1118, May 2012.

[7] B. Hassibi and H. Vikalo, “On the sphere-decoding algorithm I. expected complexity,” *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 2806–2818, Aug 2005.

[8] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, “VLSI implementation of MIMO detection using the sphere decoding algorithm,” *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 7, pp. 1566–1577, 2005.

[9] T. Xu, R. C. Grammenos, and I. Darwazeh, “FPGA implementations of real-time detectors for a spectrally efficient FDM system,” in *Telecommunications (ICT), 2013 20th International Conference on*, May 2013, pp. 1–5.

[10] R. Grammenos, “Spectrum optimisation in wireless communication systems: Technology evaluation, system design and practical implementation,” Ph.D. dissertation, University College London - Department of Electronic and Electrical Engineering, March 2013.

[11] T. Xu, R. C. Grammenos, F. Marvasti, and I. Darwazeh, “An improved fixed sphere decoder employing soft decision for the detection of non-orthogonal signals,” *Communications Letters, IEEE*, vol. 17, no. 10, pp. 1964–1967, October 2013.