Complex Block Floating-Point Format with Box Encoding For Wordlength Reduction in Communication Systems

Yeong Foong Choo*, Brian L. Evans* and Alan Gatherer†

*Wireless Networking and Communications Group, The University of Texas at Austin, Austin, TX USA

†Wireless Access Laboratory, Huawei Technologies, Plano, TX USA

*yeongfoong.choo@utexas.edu, bevans@ece.utexas.edu †alan.gatherer@huawei.com

Abstract— We propose a new complex block floating-point format to reduce implementation complexity. The new format achieves wordlength reduction by sharing an exponent across the block of samples, and uses box encoding for the shared exponent to reduce quantization error. Arithmetic operations are performed on blocks of samples at time, which can also reduce implementation complexity. For a case study of a baseband quadrature amplitude modulation (QAM) transmitter and receiver, we quantify the tradeoffs in signal quality vs. implementation complexity using the new approach to represent IQ samples. Signal quality is measured using error vector magnitude (EVM) in the receiver, and implementation complexity is measured in terms of arithmetic complexity as well as memory allocation and memory input/output rates. The primary contributions of this paper are (1) a complex block floating-point format with box encoding of the shared exponent to reduce quantization error, (2) arithmetic operations using the new complex block floating-point format, and (3) a QAM transceiver case study to quantify signal quality vs. implementation complexity tradeoffs using the new format and arithmetic operations.

Index Terms—Complex block floating-point, discrete-time baseband QAM.

Fig. 1. 32-bit equivalent SIMD ALU in Exponent Box Encoding format

I. INTRODUCTION

Energy-efficient data representation in application specific baseband transceiver hardware are in demand resulting from energy costs involved in baseband signal processing [1]. In macrocell base stations, about ten percent of energy cost contribute towards digital signal processing (DSP) modules while power amplification and cooling processes consume more than 70% of total energy [2]. The energy consumption by DSP modules relative to power amplification and cooling will increase in future designs of small cell systems because low-powered cellular radio access nodes handle a shorter radio range [2]. The design of energy-efficient number representation will reduce overall energy consumption in base stations.

In similar paper, baseband signal compression techniques have been researched for both uplink and downlink. The methods in [3], [4], and [5] suggest resampling baseband signals to Nyquist rate, block scaling, and non-linear quantization. All three papers report transport data rate gain of 3x to 5x with less than 2% EVM loss. In [5], cyclic prefix replacement technique is used to counter the effect of resampling, which would add processing overhead to the system. In [4] and [6], noise shaping technique shows improvement of in-band signal-to-noise ratio (SNR). In [7], transform coding technique 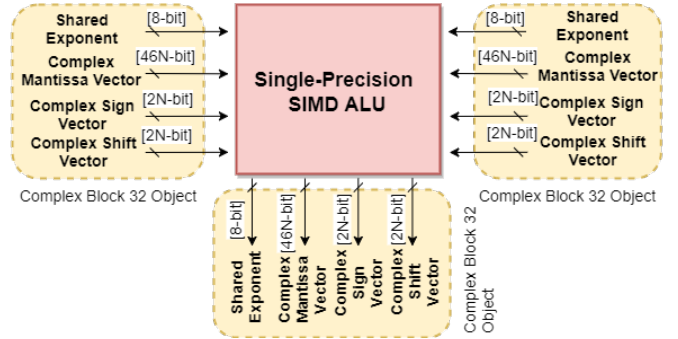is suggested for block compression of baseband signals in the settings of multiple users and multi-antenna base station. Transform coding technique reports potential of 8x transport data rate gain with less than 3% EVM loss. The above methods achieve end-to-end compression in a transport link and incur delay and energy cost for the compression and decompression at the entry and exit points, respectively. The overall energy cost reduction is not well quantified. This motivates the design of energy-efficient data representation and hardware arithmetic units with low implementation complexity.

In [8], Common Exponent Encoding is proposed to represent 32-bit complex floating-point data by only 29-bit wordlength in hardware to achieve 3-bit savings. The method in [8] shows 10% reduction of registers and memory footprints with a tradeoff of 10% increase in arithmetic units. In [9], exponential coefficient scaling is proposed to allocate 6 bits to represent real-valued floating-point data. The method in [9] achieves 37x reduction in quantization errors, 1.2x reduction in logic gates, and 1.4x reduction in energy per cycle compared to 6-bit fixed-point representation. Both papers report less than 2 dB of signal-to-quantization-noise ratio (SQNR).

Contributions: Our method applies the Common Exponent Encoding proposed by [8] and adds a proposed Exponent Box Encoding to retain high magnitude-phase resolution. This paper identifies the computational complexity of complex block addition, multiplication, and convolution and computes reference EVM on the arithmetic output. We apply the new complex block floating-point format to case study of baseband QAM transmitter chain and receiver chain. We also reduce implementation complexity in terms of memory reads/writes rates, and multiply-accumulate operations. We base the sig-

TABLE I
DEFINITION & BIT WIDTHS UNDER IEEE-754 NUMBER FORMAT [10]

Components	Definition	Bit Widths, B
Wordlength, W	N_w	$\{16, 32, 64\}$
Sign, S	N_s	$\{1\}$
Exponent, E	N_e	$\{5, 8, 11\}$
Mantissa, M	N_m	$\{10, 23, 52\}$

TABLE II
DEFINITION & BIT WIDTHS UNDER COMMON EXPONENT ENCODING [8]

Components	Definition	Bit Widths, B
Common Exponent, E	N_e	$\{5, 8, 11\}$
Real / Imaginary, $\boldsymbol{S}$	$N_s^{R,I}$	$\{1\}$
Real / Imaginary Lead, $\boldsymbol{L}$	$N_l^{R,I}$	$\{1\}$
Real / Imaginary Mantissa, $\boldsymbol{M}$	$N_m^{R,I}$	$\{10, 23, 52\}$

TABLE III
DEFINITION & BIT WIDTHS UNDER EXPONENT BOX ENCODING

Components	Definition	Bit Widths , B
Common Exponent, E	N_e	$\{5, 8, 11\}$
Real / Imaginary Sign, $\boldsymbol{S}$	$N_s^{R,I}$	$\{1\}$
Real / Imaginary Lead, $\boldsymbol{L}$	$N_l^{R,I}$	$\{1\}$
Real / Imaginary Box Shift, $\boldsymbol{X}$	$N_x^{R,I}$	$\{1\}$
Real / Imaginary Mantissa, $\boldsymbol{M}$	$N_m^{R,I}$	$\{10, 23, 52\}$

nal quality of our method on the measurement of EVM at the receiver. Our method achieves end-to-end complex block floating-point representation.

II. METHODS

This section describes the data structure used in new representation of complex block floating-point [8] and suggests a new mantissa scaling method in reducing quantization error. In IEEE 754 format, the exponents of complex-valued floating-point data are separately encoded. Common Exponent Encoding technique [8] allows common exponent sharing that has weak encoding of phase resolution.

A. Common Exponent Encoding Technique

Table I summarizes the wordlength precision of real-valued floating-point data in IEEE-754 encoding [10]. We define B_w-bit as the wordlength of scalar floating-point data. A complex-valued floating-point data requires $2B_w$-bit and a complex block floating-point of N_v samples requires $2N_vB_w$-bit.
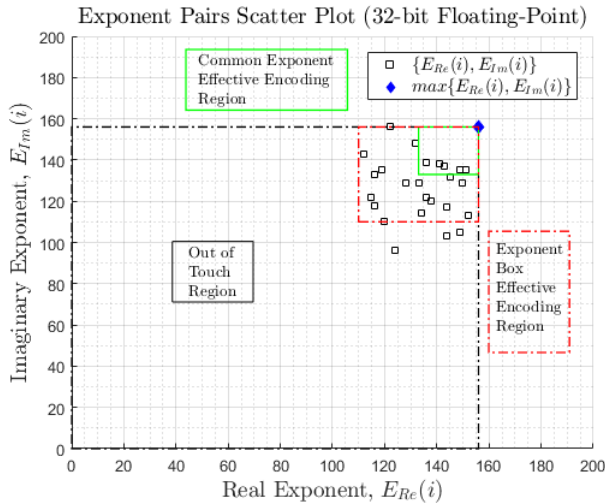
The method in [11] assumes only magnitude correlation in

Fig. 2. Scatter plot of $N = 25$ complex-valued exponent pairs $X \sim N(130, 12^2)$ (□) and potential candidate for common exponent (◆)

the oversampled complex block floating-point data. This assumption allows common exponent be jointly encoded across complex block floating-point of N_v samples defined in Table II. The implied leading bit of 1 of each floating-point data is first uncovered. The common exponent is selected from the largest unsigned exponent across the complex block. All mantissa values are successively scaled down by the difference between common exponent and its original exponent. Therefore, each floating-point data with smaller exponents value loses leading bit of 1. The leading bit of complex block floating-point is explicitly coded as N_l, using B_l-bit. The sign bits are left unchanged. A complex block floating-point of N_v samples requires $\{2N_v(B_s + B_l + B_m) + B_e\}$-bit.

We derive the maximum allowed exponent difference under Common Exponent Encoding in Appendix . Mantissa values could be reduced to zero as a result of large phase difference. Figure 2 shows the Effective Encoding Region (EER) under Common Exponent Encoding technique (□). Exponent pairs outside the EER will have corresponding mantissa values reduce to zero.

B. Exponent Box Encoding Technique

The Common Exponent Encoding technique suffers high quantization and phase error in the complex block floating-point of high dynamic range. Exponent Box Encoding is suggested to reduce quantization error of complex-valued floating-point pairs by allocating $2N_v$-bit per complex block. Figure 2 shows the Effective Encoding Region under Exponent Box Encoding technique (□) which has four times larger the area of EER of Common Exponent Encoding technique (□).

The use of 2-bit per complex sample replaces the mantissas rescaling operation with exponents addition/ subtraction. We are able to preserve more leading bits of mantissas values which improve the accuracy of complex block multiplication and complex block convolution results. A complex block floating-point of N_v samples requires $\{2N_v(B_s + B_l + B_x + B_m) + B_e\}$-bit.

Arithmetic Logic Unit (ALU) hardware is designed to perform Single-Instruction Multiple-Data (SIMD) operation on complex block floating-point data. The Exponent Box Encoding is performed when converting to Exponent Box Encoding format. The Exponent Box Decoding is performed at the pre-processing of mantissas in Complex Block Addition and pre-processing of exponents in Complex Block Multiply.

Table IV summarizes the wordlength analysis required by complex block floating-point of B_v samples. The Exponent

TABLE IV
WORDLENGTH REQUIREMENT BY N_v COMPLEX-VALUED SAMPLES

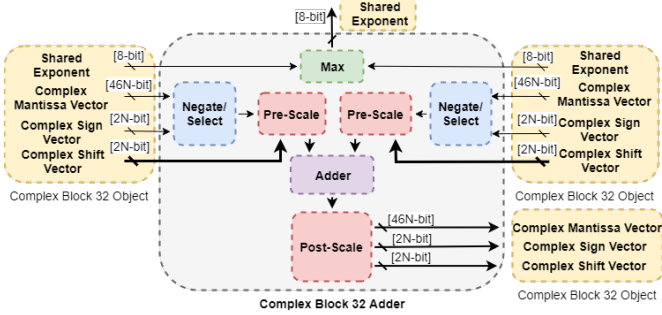| Encoding	Bit Widths
Complex IEEE754	$2N_v(B_s + B_e + B_m)$
Common Exponent	$2N_v(B_s + B_l + B_m) + B_e$
Exponent Box	$2N_v(B_s + B_l + B_x + B_m) + B_e$

Fig. 3. Pre-Scale mantissas in Complex Block Add

Box Encoding and Exponent Box Decoding algorithms are described as follows:

Algorithm 1 Exponent Box Encoding

$Let \quad U \leftarrow max\{E\{s\}\} - B_m$
for $i^{th} \in N_v \quad \{R/I\} \, samples$ **do**
$\quad$ **if** $N_e^R\{i\} < U$ **then**
$\quad\quad N_e^R\{i\} \leftarrow N_e^R\{i\} + B_m$
$\quad\quad N_x^R\{i\} \leftarrow 1$
$\quad$ **if** $N_e^I\{i\} < U$ **then**
$\quad\quad N_e^I\{i\} \leftarrow N_e^I\{i\} + B_m$
$\quad\quad N_x^I\{i\} \leftarrow 1$

Algorithm 2 Exponent Box Decoding

for $i^{th} \in N_v \quad \{R/I\} \, samples$ **do**
$\quad$ **if** $N_x^R\{i\} \equiv 1$ **then**
$\quad\quad N_e^R\{i\} \leftarrow N_e^R\{i\} - B_m$
$\quad$ **if** $N_x^I\{i\} \equiv 1$ **then**
$\quad\quad N_e^I\{i\} \leftarrow N_e^I\{i\} - B_m$

III. ARITHMETIC UNIT

We identify the arithmetic units predominantly used on complex block floating-point data. Complex-valued multiplication and addition are two primary ALU required in convolution operation. This section identifies the complexity of pre-processing and post-processing mantissas and exponents in the complex block addition, multiplication, and convolution arithmetic. Table V describes the worst-case complexity analysis of complex block ALU on encoding format described in Section .
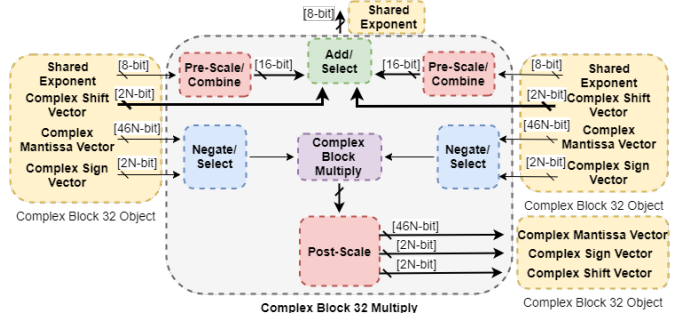
A. Complex Block Addition

Fig. 4. Pre-Scale exponents in Complex Block Multiply

Figure 3 shows simplified block diagram for Complex Block Addition. Let $X_1, X_2, Y \in \mathbb{C}^{1 \times N}$ be complex-valued row vectors, such that,

$$\Re\{Y\} = \Re\{X_1\} + \Re\{X_2\}$$
$$\Im\{Y\} = \Im\{X_1\} + \Im\{X_2\} \quad (1)$$

In IEEE-754 encoding format, complex block addition is implemented as two real-valued addition. There are four exponents to the two complex inputs and two exponents to the complex output. Each real-valued addition block requires one mantissa pre-scaling, one mantissa post-scaling, and one exponent arithmetic. Therefore, complex block addition requires two mantissas pre-scaling, two mantissas post-scaling, and two exponents arithmetic per sample.

In Common Exponent and Exponent Box Encoding, there are two shared exponents to the two complex block inputs and one shared exponent to the complex block output. Complexity on shared exponent arithmetic is $O(1)$. We pre-scale the mantissas corresponding to the smaller exponent and post-scale the mantissas of the complex block output. With Exponent Box Encoding in the worst case, we require two mantissas pre-scaling and one mantissas post-scaling.

B. Complex Block Multiplication

Figure 4 shows simplified block diagram for Complex Block Multiplication. Let $X_1, X_2, Y \in \mathbb{C}^{1 \times N}$ be complex-valued row vectors, where $\bullet$ denotes element-wise multiply, such that,

$$\Re\{Y\} = \Re\{X_1\} \bullet \Re\{X_2\} - \Im\{X_1\} \bullet \Im\{X_2\}$$
$$\Im\{Y\} = \Re\{X_1\} \bullet \Im\{X_2\} + \Im\{X_1\} \bullet \Re\{X_2\} \quad (2)$$

In IEEE-754 encoding format, complex block multiplication is implemented as four real-valued multiplication and two real-valued addition. Each real-valued multiplication requires one mantissa post-scaling and one exponent arithmetic. Each real-valued addition requires one mantissa pre-scaling, one mantissa post-scaling, and one exponent arithmetic. Complex block multiply requires two mantissas pre-scaling, six mantissas post-scaling, and six exponent arithmetic per sample.

In Common Exponent and Exponent Box Encoding, we need two exponent arithmetic for multiply and normalization

Block Addition	Mantissas Scaling	Exponents Arithmetic
Complex IEEE754	$4 * N$	$2 * N$
Common Exponent	$4 * N$	2
Exponent Box	$8 * N$	4
Block Multiplication	**Mantissas Scaling**	**Exponents Arithmetic**
Complex IEEE754	$8 * N$	$6 * N$
Common Exponent	$8 * N$	2
Exponent Box	$16 * N$	5
Convolution	**Mantissas Scaling**	**Exponents Arithmetic**
Complex IEEE754	$6 * N_1 N_2 + 4 * (N_1 - 1)(N_2 - 1)$	$6 * N_1 N_2 + 2 * (N_1 - 1)(N_2 - 1)$
Common Exponent	$6 * N_1 N_2 + 4 * (N_1 - 1)(N_2 - 1)$	$3 * (N_1 + N_2 - 1) + 1$
Exponent Box	$10 * N_1 N_2 + 8 * (N_1 - 1)(N_2 - 1)$	$3 * (N_1 + N_2 - 1) + 1$

of the complex block output. With Exponent Box Encoding in the worst case, we need eight more mantissas post-scaling. Also, the Shift Vectors allow for four possible intermediate exponent values instead of one intermediate exponent value in Common Exponent Encoding.

C. Complex Convolution

Let $\boldsymbol{X_1} \in \mathbb{C}^{1 \times N_1}$, $\boldsymbol{X_2} \in \mathbb{C}^{1 \times N_2}$, and $\boldsymbol{Y} \in \mathbb{C}^{1 \times (N_1 + N_2 - 1)}$ be complex-valued row vectors, where $*$ denotes convolution, such that,

$$\Re\{\boldsymbol{Y}\} = \Re\{\boldsymbol{X_1} * \boldsymbol{X_2}\}$$
$$\Im\{\boldsymbol{Y}\} = \Im\{\boldsymbol{X_1} * \boldsymbol{X_2}\} \tag{3}$$

We assume $N_1 < N_2$ for practical reason where the model of channel impulse response has shorter sequence than the discrete-time samples. Each term in the complex block output is complex inner product of two complex block input of varying length between 1 and $min\{N_1, N_2\}$. Complex convolution is implemented as complex block multiplication and accumulation of intermediate results. We derive the processing complexity of mantissas and exponents in Appendix .

IV. SYSTEM MODEL

We apply Exponent Box Encoding to represent IQ components in baseband QAM transmitter in Figure 5 and baseband QAM receiver in Figure 6. The simulated channel model is Additive White Gaussian Noise (AWGN). Table VI contains the parameter definitions and values used in MATLAB simulation and Table VII summarizes the memory input/output rates (bits/sec) and multiply-accumulate rates required by discrete-time complex QAM transmitter and receiver chains.

A. Discrete-time Complex Baseband QAM Transmitter

We encode complex block IQ samples in Exponent Box Encoding and retain the floating-point resolution in 32-bit IEEE-754 precision in our model. For simplicity, we select block size to be, $N_v = L^{TX} f_{sym}$. The symbol mapper generates a $L^{TX} f_{sym}$-size of complex block IQ samples that shares common exponent. Pulse shape filter is implemented as Finite Impulse Response (FIR) filter of N^{TX}-order and requires complex convolution on the upsampled complex block IQ samples.

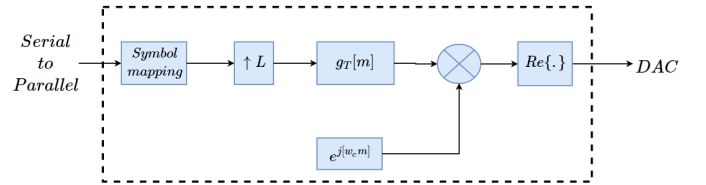| QAM Parameters	Definition	Values / Types
Constellation Order	M	1024
Transceiver Parameters	**Definition**	**Values / Types**
Up-sample Factor	L^{TX}, L^{RX}	4
Symbol Rate (Hz)	f_{sym}	2400
Filter Order	N^{TX}, N^{RX}	32th
Pulse Shape	g^{TX}, g^{RX}	Root-Raised Cosine
Excess Bandwidth Factor	α^{Tx}, α^{RX}	0.2

Fig. 5. Block diagram of discrete-time complex baseband QAM transmitter

B. Discrete-time Complex Baseband QAM Receiver

Due to the channel effect such as fading in practice, the received signals will have larger span in magnitude-phase response. The Common Exponent Encoding applied on sampled complex block IQ samples is limited to selecting window size of minimum phase difference. The Common Exponent Encoding must update its block size at the update rate of gain by the Automatic Gain Control (AGC). Instead, our Exponent Box Encoding could lift the constraint and selects fixed block size, $N_v = L^{RX} f_{sym}$ in this simulation. We simulate matched filter of N^{RX}-order.
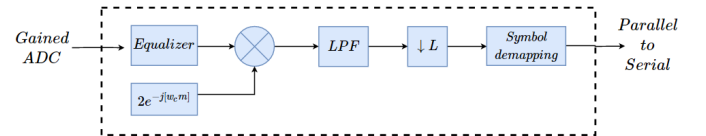
V. SIMULATION RESULTS

Fig. 6. Block diagram of discrete-time complex baseband QAM receiver

| Transmitter Chain	Memory Reads Rate (bits/sec)	Memory Writes Rate (bits/sec)	MACs / sec
Symbol Mapper	Jf_{sym}	$2f_{sym}(N_w + N_l + N_b - N_e) + N_e$	0
Upsampler	$2f_{sym}(N_w + N_l + N_b - N_e) + N_e$	$2L^{Tx}f_{sym}(N_w + N_l + N_b - N_e) + N_e$	0
Pulse Shape Filter	$(3L^{Tx}N_g^{Tx} + 1)(L^{Tx}f_{sym})(N_w + N_l + N_b - N_e) + 2N_e$	$2L^{Tx}f_{sym}(N_w + N_l + N_b - N_e) + N_e$	$(L^{Tx})^2N_g^{Tx}f_{sym}$
Receiver Chain	Memory Reads Rate (bits/sec)	Memory Writes Rate (bits/sec)	MACs / sec
Matched Filter	$(3L^{Rx}N_g^{Rx} + 1)(L^{Rx}f_{sym})(N_w + N_l + N_b - N_e) + 2N_e$	$2L^{Rx}f_{sym}(N_w + N_l + N_b - N_e) + N_e$	$(L^{Rx})^2N_g^{Rx}f_{sym}$
Downsampler	$2L^{Rx}f_{sym}(N_w + N_l - N_e) + N_e + (N_w + N_l + N_b)$	$2f_{sym}(N_w + N_l + N_b - N_e) + N_e$	0
Symbol Demapper	$2f_{sym}(N_w + N_l - N_e) + N_e + \frac{J}{2}(N_w + N_l)$	Jf_{sym}	0
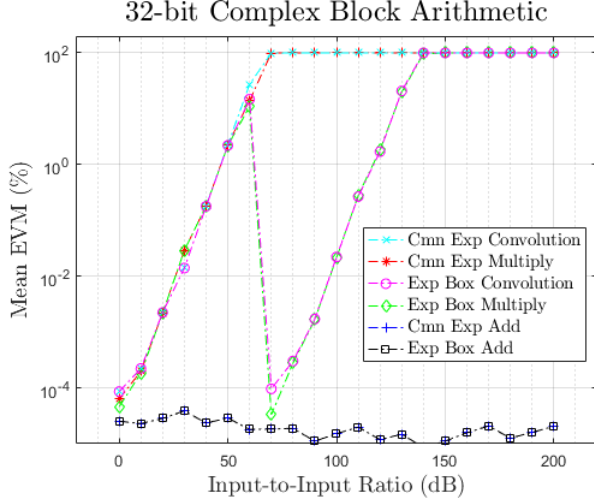
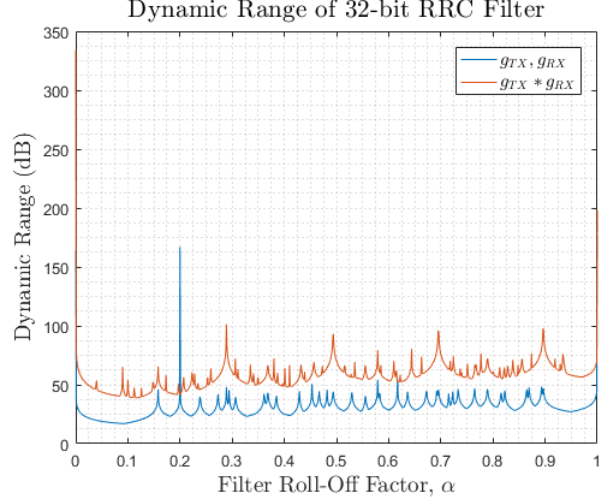Fig. 7. Error vector magnitude of 32-bit complex block arithmetic

Fig. 8. Dynamic range of 32-bit RRC filter impulse response as function of roll-off factor

A. Error Vector Magnitude on Complex Block (32-bit) ALU

Let $X, \bar{X} \in \mathbb{C}^{1 \times N}$ be complex-valued row vectors, such that X is the reference results in IEEE-754 Encoding and $\bar{X}$ is the simulated results in Complex Block Encoding.

The signal quality is measured on the complex block arithmetic results. We truncate the arithmetic results to 32-bit precision to make fair comparison. We use the Root-Mean-Squared (RMS) EVM measurement as described in the following, with $\| \bullet \|_2$ as the Euclidean Norm,

$$EVM = \frac{\| X - \bar{X} \|_2}{\| X \|_2} * 100 \qquad (4)$$

Figure 7 shows the EVM of complex block arithmetic in Section on Inputs Ratio $\in (0, 200)$ dB. In complex block addition, the Exponent Box Encoding does not show significant advantage over Common Exponent Encoding because the mantissas addition emphasizes on magnitude over phase. In complex block multiplication and convolution, the Exponent Box Encoding achieves significant reduction in encoding error over Common Exponent Encoding particularly on Inputs Ratio $\in (70, 140)$ dB where the improvement is between $(0, 99.999)\%$.

B. Error Vector Magnitude on Single-Carrier Transceiver

Figure 8 shows the dynamic range of Root-Raised Cosine (RRC) filter at transmitter and receiver and overall pulse shape response as a function of α. Figure 9 shows the EVM introduced by Complex Block Encoding under system model defined in Section . The EVM plot is indistinguishable between IEEE-754 Encoding and Complex Block Encoding. The reasons are the selection of RRC Roll-off factor and energy-normalized constellation map.

VI. CONCLUSION

Our work has identified the processing overhead of the mantissas and shared exponent in complex block floating-point arithmetic. The common exponent encoding would slightly lower the overhead in complex-valued arithmetic. The box encoding of the shared exponent gives the same quantization errors as common exponent encoding in our case study, which is a 32-bit complex baseband transmitter and receiver. Our work has also quantified memory read/write rates and multiply-accumulate rates in our case study. Future work could extend a similar approach to representing and processing IQ samples in multi-carrier and multi-antenna communication systems.
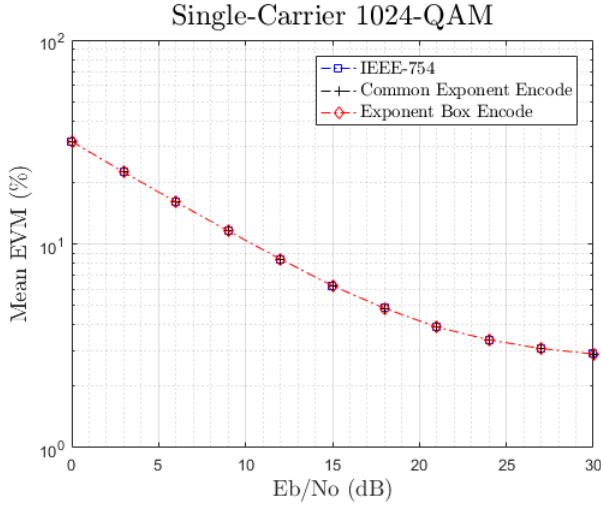
Fig. 9. Error vector magnitude between encoding techniques on complex-valued IQ samples

APPENDIX A
DERIVATION OF MAXIMUM EXPONENT DIFFERENCE UNDER COMMON EXPONENT ENCODING TECHNIQUE

Let i, j be two bounded positive real numbers, representable in floating point precision. Assume that i has larger magnitude than j, $|j| < |i|$. Define $E\{k\}$ as exponent and $M\{k\}$ as mantissa to k, and $F(k) = 2^{E\{k\}-1} - 1$ as exponent offset, where $k = \{i, j\}$. Let $E\{\Delta\}$ be the difference between two exponents, $(E\{i\} - E\{j\}) > 0$.

$$
\begin{aligned}
j &< i \\
(1.M\{j\} * 2^{E\{j\}-F(j)}) &< (1.M\{i\} * 2^{E\{i\}-F(i)}) \\
(1.M\{j\} * 2^{E\{j\}}) &< (1.M\{i\} * 2^{E\{i\}}) \\
(1.M\{j\} * 2^{E\{j\}-E\{i\}+E\{i\}}) &< (1.M\{i\} * 2^{E\{i\}}) \\
(1.M\{j\} * 2^{E\{j\}-E\{i\}}) &< (1.M\{i\}) \\
(1.M\{j\} * 2^{-E\{\Delta\}}) &< (1.M\{i\}) \\
(0.M\{j'\}) &< (1.M\{i\}) \\
where \quad M\{j'\} &= \frac{1.M\{j\}}{2^{E\{\Delta\}}}
\end{aligned}
\tag{5}
$$

The mantissa bits in $M(j')$ are truncated in practice, therefore, $E\{\Delta\}$ must be less than $M(j)$. The quantization error is the largest when the $M(j')$ gets zero when $M(j)$ is nonzero.

APPENDIX B
DERIVATION OF PRE / POST PROCESSING COMPLEXITY OF COMPLEX-VALUED CONVOLUTION

Let $N_{mult}^{mant}, N_{add}^{mant}, N_{mult}^{exp}, N_{add}^{exp}$ be processing complexity of mantissas and exponents determined in Section .

Among the first and last N_1 terms of $\boldsymbol{Y}$, they are computed by complex inner product of $i \in \{1, ..., N_1\}$ input terms from $\boldsymbol{X_1}, \boldsymbol{X_2}$ and requires $\frac{(N_1)(N_1+1)}{2}(N_{mult})$ and $\frac{(N_1-1)(N_1)}{2}(N_{add})$. Among the centering $N_2 - N_1$ terms of $\boldsymbol{Y}$, they are computed by complex inner product of N_1 input

terms from $\boldsymbol{X_1}, \boldsymbol{X_2}$ and requires $(N_2 - N_1)((N_1)(N_{mult}) + (N_1 - 1)(N_{add}))$.

Overall Multiplication Requirement (N_{mult}):

$$
\begin{aligned}
&\frac{1}{2}(N_1)(N_1 + 1) + (N_2 - N_1)(N_1) + \frac{1}{2}(N_1 - 1)(N_1) \\
&= \frac{1}{2}(N_1^2 + N_1) + (N_2 N_1 - N_1^2) + \frac{1}{2}(N_1^2 - N_1) \\
&= \frac{1}{2}(2N_1^2) + (N_2 N_1 - N_1^2) \\
&= N_1^2 + (N_2 N_1 - N_1^2) \\
&= N_2 N_1
\end{aligned}
\tag{6}
$$

Overall Addition Requirement (N_{add}):

$$
\begin{aligned}
&\frac{1}{2}(N_1 - 1)(N_1) + (N_2 - N_1)(N_1 - 1) + \frac{1}{2}(N_1 - 2)(N_1 - 1) \\
&= \frac{1}{2}(N_1^2 - N_1 + N_1^2 - 3N_1 + 2) + (N_2 - N_1)(N_1 - 1) \\
&= (N_1^2 - 2N_1 + 1) + (N_2 - N_1)(N_1 - 1) \\
&= (N_1 - 1)(N_1 - 1) + (N_2 - N_1)(N_1 - 1) \\
&= (N_1 - 1)(N_1 - 1 + N_2 - N_1) \\
&= (N_1 - 1)(N_2 - 1)
\end{aligned}
\tag{7}
$$

Mantissa processing requirement is $(N_{mult}^{mant})(N_2 N_1) + (N_{add}^{mant})(N_1 - 1)(N_2 - 1)$ and exponent processing requirement is $(N_{mult}^{exp})(N_2 N_1) + (N_{add}^{exp})(N_1 - 1)(N_2 - 1)$.

REFERENCES

[1] G. Fettweis and E. Zimmermann, "ICT energy consumption-trends and challenges," in *Proc. Int. Symposium on Wireless Personal Multimedia Communications*, vol. 2, no. 4, 2008, p. 6.

[2] O. Blume, D. Zeller, and U. Barth, "Approaches to energy efficient wireless access networks," in *Int. Symposium on Communications, Control and Signal Processing*, March 2010, pp. 1–5.

[3] D. Samardzija, J. Pastalan, M. MacDonald, S. Walker, and R. Valenzuela, "Compressed Transport of Baseband Signals in Radio Access Networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 9, pp. 3216–3225, September 2012.

[4] K. F. Nieman and B. L. Evans, "Time-domain compression of complex-baseband LTE signals for cloud radio access networks," in *Proc. IEEE Global Conference on Signal and Information Processing*, Dec 2013, pp. 1198–1201.

[5] D. Peng-ren and Z. Can, "Compressed transport of baseband signals in cloud radio access networks," in *Proc. Int. Conf. Communications and Networking in China (CHINACOM)*, Aug 2014, pp. 484–489.

[6] L. S. Wong, G. E. Allen, and B. L. Evans, "Sonar data compression using non-uniform quantization and noise shaping," in *Asilomar Conference on Signals, Systems and Computers*, Nov 2014, pp. 1895–1899.

[7] J. Choi, B. L. Evans, and A. Gatherer, "Space-time fronthaul compression of complex baseband uplink LTE signals," in *Proc. IEEE Int. Conference on Communications*, May 2016, pp. 1–6.

[8] N. Cohen and S. Weiss, "Complex Floating Point A Novel Data Word Representation for DSP Processors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 10, pp. 2252–2262, Oct 2012.

[9] Z. Wang, J. Zhang, and N. Verma, "Reducing quantization error in low-energy FIR filter accelerators," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, April 2015, pp. 1032–1036.

[10] "IEEE Standard for Floating-Point Arithmetic," *IEEE Std 754-2008*, pp. 1–70, Aug 2008.

[11] N. McGowan, B. Morris, and E. Mah, "Compact floating point delta encoding for complex data," Mar. 3 2015, US Patent 8,972,359. [Online]. Available: https://www.google.com/patents/US8972359