VLSI Design of a 3-bit Constant-Modulus Precoder for Massive MU-MIMO

Oscar Castañeda[1], Sven Jacobsson[1,2,3], Giuseppe Durisi[2], Tom Goldstein[4], and Christoph Studer[1]

[1]Cornell University, Ithaca, NY; oc66@cornell.edu; studer@cornell.edu; web: http://vip.ece.cornell.edu
[2]Ericsson Research, Gothenburg, Sweden; sven.jacobsson@ericsson.com
[3]Chalmers University of Technology, Gothenburg, Sweden; durisi@chalmers.se
[4]University of Maryland, College Park, MD; tomg@cs.umd.edu

Abstract—**Fifth-generation (5G) cellular systems will build on massive multi-user (MU) multiple-input multiple-output (MIMO) technology to attain high spectral efficiency. However, having hundreds of antennas and radio-frequency (RF) chains at the base station (BS) entails prohibitively high hardware costs and power consumption. This paper proposes a novel nonlinear precoding algorithm for the massive MU-MIMO downlink in which each RF chain contains an 8-phase (3-bit) constant-modulus transmitter, enabling the use of low-cost and power-efficient analog hardware. We present a high-throughput VLSI architecture and show implementation results on a Xilinx Virtex-7 FPGA. Compared to a recently-reported nonlinear precoder for BS designs that use two 1-bit digital-to-analog converters per RF chain, our design enables up to 3.75 dB transmit power reduction at no more than a $2.7\times$ increase in FPGA resources.**

I. INTRODUCTION

Fifth-generation (5G) cellular communication systems are widely expected to rely on massive multi-user (MU) multiple-input multiple-output (MIMO) technology to achieve significant improvements in spectral efficiency compared to existing small-scale MIMO systems [2]–[4]. MU-MIMO equips the base station (BS) with hundreds of antennas and radio-frequency (RF) chains, enabling one to simultaneously serve tens of user equipments (UEs) in the same time-frequency resource via fine-grained beamforming. Unfortunately, scaling conventional multi-antenna BS architectures (that use high-precision RF chains) to BSs with hundreds of antenna elements entails a significant increase in system costs and circuit power consumption. Hence, to make massive MU-MIMO systems inexpensive and power-efficient, novel BS architectures and suitable baseband-processing algorithms are necessary.

Low-Precision BS Architechures: The use of low-precision digital-to-analog-converters (DACs) at the BS in the massive

MU-MIMO downlink enables significant reductions in terms of system costs and circuit power consumption. The key challenge with such low-precision BS architectures is to maintain high spectral efficiency, which requires sophisticated baseband-processing algorithms. While linear precoders, e.g., maximal-ratio transmission (MRT) and zero-forcing (ZF), followed by quantization exhibit low complexity [5]–[7], sophisticated non-linear precoders can achieve superior performance, especially for the extreme case of using only a pair of 1-bit DACs per RF chain [8]–[13]. Recently, reference [1] presented VLSI designs of nonlinear precoders for systems that use a pair of 1-bit DACs per RF chain, which demonstrates that nonlinear precoding is feasible in practice for such low-precision BS architectures.

The use of 1-bit DACs at the BS ensures that the precoded signal has constant-modulus (CM), i.e., the precoded signal's amplitude is equal on all antennas and constant over time, which enables the use of low-cost and power-efficient analog circuitry, such as nonlinear power amplifiers. Recently, nonlin-ear precoders for 8-phase (3-bit) CM transmitters, i.e., the setup considered in this work, were proposed in [14], [15]. It remains, however, an open question whether the algorithms proposed in [14], [15] can be implemented efficiently in hardware.

Contributions: This paper develops a novel nonlinear pre-coding algorithm in which each RF chain contains an 8-phase (3-bit) CM transmitter that enables efficient analog circuitry while surpassing the error-rate performance of systems that use a pair of 1-bit DACs (i.e., 4 phases) per RF chain. We propose a nonconvex algorithm to solve the associated 8-phase (3-bit) CM precoding problem in an efficient manner, and we develop a VLSI architecture that uses a fast matrix-vector multiplication engine based on Cannon's algorithm [16]. We show Xilinx Virtex-7 FPGA implementation results and provide a comparison with the C2PO precoder proposed in [1].

II. SYSTEM MODEL AND CM PRECODING

A. System Model

We consider the single-cell, narrowband massive MU-MIMO downlink system shown in Fig. 1. Here, the BS, which is equipped with B antennas, serves $U \leq B$ single-antenna UEs. The narrowband downlink channel is modeled by $\mathbf{y} = \mathbf{Hx} + \mathbf{n}$, where $\mathbf{y} = [y_1, \ldots, y_U]^T \in \mathbb{C}^U$ contains the received signals at all UEs, $\mathbf{H} \in \mathbb{C}^{U \times B}$ is the channel matrix (which we assume is
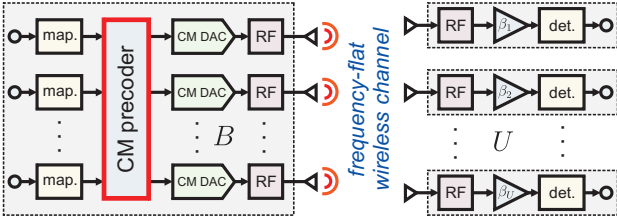
Fig. 1. Overview of the considered massive MU-MIMO downlink system with CM DACs. Left: B antenna massive MU-MIMO BS containing a CM precoder that mitigates multi-user interference and quantization artifacts in the CM DACs; Right: U single-antenna UEs.

Fig. 2. Left: 8-phase CM alphabet (convex polytope in blue); right: projection regions within the first quadrant of the 8-phase CM alphabet.

known to the BS), $\mathbf{n} \in \mathbb{C}^U$ models i.i.d. circularly-symmetric complex Gaussian noise with variance N_0 per complex entry, and $\mathbf{x} \in \mathcal{X}^B$ is the so-called *precoded vector*, where $\mathcal{X}$ is the transmit alphabet. In this work, we require that $\mathcal{X}$ has finite cardinality and that the entries of $\mathcal{X}$ have CM. Specifically, the CM alphabet is $\mathcal{X} = \{\exp(j2\pi p/P) \mid p = 0, \dots, P-1\}$ where P denotes the number of phases and $\log_2(P)$ the number of bits per RF chain. The CM constraint ensures that $\|\mathbf{x}\|_2^2 = B$.

B. Constant-Modulus (CM) Precoding

The precoder at the BS maps the symbol vector $\mathbf{s} = [s_1, \dots, s_U]^T$ into the precoded vector $\mathbf{x} \in \mathcal{X}^B$. Here, $s_u \in \mathcal{O}$ is the constellation point intended for the uth UE ($u = 1, \dots, U$), where $\mathcal{O}$ is the constellation set (e.g., 16-QAM). We assume that each UE $u = 1, \dots, U$ rescales its received signal y_u by a factor $\beta_u \in \mathbb{C}$ to compute an estimate $\hat{s}_u = \beta_u y_u$ of the transmitted symbol s_u. Nonlinear precoders that minimize the mean-squared error (MSE) between the transmitted and the estimated symbols solve the following optimal precoding problem (OPP) [1]:

$$(\text{OPP}) \quad \{\hat{\mathbf{x}}, \hat{\beta}\} = \underset{\mathbf{x} \in \mathcal{X}^B, \beta \in \mathbb{C}}{\arg\min} \|\mathbf{s} - \beta\mathbf{H}\mathbf{x}\|_2^2 + |\beta|^2 U N_0.$$

Here, we assume that $\beta = \beta_u$ for $u = 1, \dots, U$; as shown in [9], the UEs are able to accurately learn $\hat{\beta}$. For systems that use a pair of 1-bit DACs per RF chain ($P = 4$ phases), methods that solve (OPP) approximately using convex [8], [9] and nonconvex [1] relaxation have been proposed recently. In what follows, we present a novel precoder specifically designed for CM transmitters with 3 bits per RF chain ($P = 8$ phases), which enables significant error-rate performance improvements compared to systems with 2 bits per RF chain ($P = 4$ phases), without requiring complex RF circuitry.

III. C3PO: CONSTANT-MODULUS 3-BIT PRECODING

A. Relaxing the Problem (OPP)

To find an approximate solution to (OPP) via methods that can be implemented efficiently, we perform the following approximations. First, we let $N_0 \to 0$, i.e., we assume that the system operates in the high-SNR regime. Then, we use the following approximation [1, Eq. (2)]:

$$\min_{\mathbf{x} \in \mathcal{X}^B} \min_{\beta \in \mathbb{C}} \|\mathbf{s} - \beta\mathbf{H}\mathbf{x}\|_2^2 \approx \min_{\mathbf{x} \in \mathcal{X}^B} \min_{\alpha \in \mathbb{C}} \|\alpha\mathbf{s} - \mathbf{H}\mathbf{x}\|_2^2.$$
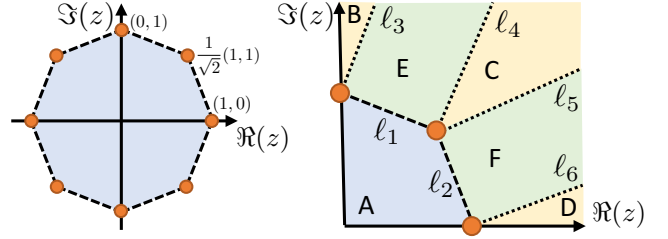
These two approximations result in the following problem:

$$(\text{OPP}^*) \quad \{\hat{\mathbf{x}}, \hat{\alpha}\} = \underset{\mathbf{x} \in \mathcal{X}^B, \alpha \in \mathbb{C}}{\arg\min} \|\alpha\mathbf{s} - \mathbf{H}\mathbf{x}\|_2^2.$$

We next compute $\hat{\alpha}$ by minimizing the objective function of (OPP*), which results in $\hat{\alpha} = \mathbf{s}^H \mathbf{H}\mathbf{x}/\|\mathbf{s}\|_2^2$. Substituting $\hat{\alpha}$ in (OPP*) yields $\|\hat{\alpha}(\mathbf{x})\mathbf{s} - \mathbf{H}\mathbf{x}\|_2^2 = \|\mathbf{A}\mathbf{x}\|_2^2$ with $\mathbf{A} = \mathbf{Q}\mathbf{H}$ and $\mathbf{Q} = \mathbf{I}_U - \mathbf{s}\mathbf{s}^H/\|\mathbf{s}\|_2^2$. Hence, we can simplify (OPP*) as

$$(\text{OPP}^{**}) \quad \hat{\mathbf{x}} = \underset{\mathbf{x} \in \mathcal{X}^B}{\arg\min} \tfrac{1}{2}\|\mathbf{A}\mathbf{x}\|_2^2.$$

The factor $1/2$ does not affect the solution of (OPP**). We now replace the finite-phase constraint $\mathbf{x} \in \mathcal{X}^B$ by the convex polytope surrounding the points $\mathcal{X} = \{\mathsf{x}_p\}_{p=1}^P$ given by

$$\mathcal{B} = \left\{ \textstyle\sum_{p=1}^P \alpha_p \mathsf{x}_p \mid (\alpha_p \geq 0, \forall p) \wedge \sum_{p=1}^P \alpha_p = 1 \right\}.$$

For 3-bit CM precoding, the boundary of the convex polytope $\mathcal{B}$ is a regular octagon (see Fig. 2). Unfortunately, solving (OPP**) over the relaxed set $\mathbf{x} \in \mathcal{B}^B$ yields the all-zeros vector. We therefore attempt to solve the following modified problem via forward-backward splitting (FBS) [17]–[19]:

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in \mathcal{B}^B}{\arg\min} \ \tfrac{1}{2}\|\mathbf{A}\mathbf{x}\|_2^2 - \tfrac{\delta}{2}\|\mathbf{x}\|_2^2, \tag{1}$$

where the concave regularizer $-\frac{\delta}{2}\|\mathbf{x}\|_2^2$ with $\delta > 0$ forces the solution $\hat{\mathbf{x}}$ to lie at the boundary of the convex polytope $\mathcal{B}^B$. As the problem in (1) is nonconvex, FBS is not guaranteed to converge to an optimal solution. Nevertheless, the algorithm proposed exhibits good empirical performance (see Sec. IV).

B. The C3PO Algorithm

FBS is an efficient numerical method to solve convex optimization problems whose objective function can be decomposed as $f(\mathbf{x}) + g(\mathbf{x})$, where the function f is smooth and convex, and the function g is convex but not necessarily smooth or bounded. FBS consists of the following iteration [17], [18]:

$$\mathbf{x}^{(t+1)} = \text{prox}_g\big(\mathbf{z}^{(t+1)}; \tau^{(t)}\big) \text{ with } \mathbf{z}^{(t+1)} = \mathbf{x}^{(t)} - \tau^{(t)}\nabla f(\mathbf{x}^{(t)})$$

for $t = 1, 2, \dots, t_{\max}$ or until convergence. Here, the sequence $\{\tau^{(t)} > 0\}$ contains suitably chosen step-size parameters and $\nabla f(\mathbf{x})$ is the gradient of the smooth function f, and the so-called *proximal operator* for the function g is defined by [20]

$$\text{prox}_g(\mathbf{z}; \tau) = \underset{\mathbf{x} \in \mathbb{C}^B}{\arg\min} \left\{ \tau g(\mathbf{x}) + \tfrac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 \right\}.$$

To approximately solve (1) using FBS, we set

$$f(\mathbf{x}) = \tfrac{1}{2}\|\mathbf{A}\mathbf{x}\|_2^2 \ \text{ and } \ g(\mathbf{x}) = \chi\big(\mathbf{x} \in \mathcal{B}^B\big) - \tfrac{\delta}{2}\|\mathbf{x}\|_2^2,$$

where χ is a characteristic function that is zero if $\mathbf{x} \in \mathcal{B}^B$ and infinity otherwise. For these choices, the gradient is given by $\nabla f(\mathbf{x}) = \mathbf{A}^H \mathbf{A}\mathbf{x}$ and the proximal operator is detailed in Sec. III-C. Furthermore, we use a constant step size $\tau = \tau^{(t)}$. The resulting algorithm is as follows:

Algorithm 1 (C3PO). *Initialize* $\mathbf{x}^{(1)} = \mathbf{H}^H\mathbf{s}$ *and fix the parameters* δ *and* τ *so that* $\tau\delta < 1$. *Then, for every iteration* $t = 1, 2, \ldots, t_{max}$ *compute:*

$$\mathbf{z}^{(t+1)} = \mathbf{x}^{(t)} - \tau\mathbf{A}^H\mathbf{A}\mathbf{x}^{(t)} \qquad (2)$$

$$\mathbf{x}^{(t+1)} = \mathrm{prox}_g(\mathbf{z}^{(t+1)}; \tau). \qquad (3)$$

The prox_g *operator is applied element-wise to* $\mathbf{z}^{(t+1)}$ *and detailed in Sec. III-C. In the last iteration* t_{max}, *the output* $\mathbf{x}^{(t_{max}+1)}$ *is quantized to the 3-bit CM alphabet* $\mathcal{X}^B$.

The most costly operation of C3PO is the matrix-vector product in step (2), which we compute as: $\mathbf{A}^H\mathbf{A} = \mathbf{H}^H\mathbf{H} - \mathbf{v}\mathbf{v}^H = \overline{\mathbf{H}}^\Upsilon\overline{\mathbf{H}}$, where $\mathbf{v} = \mathbf{H}^H\mathbf{s}/\|\mathbf{s}\|_2$ is a normalized version of the MRT vector; the augmented matrices $\overline{\mathbf{H}} = [\mathbf{H}; \mathbf{v}^H]$ and $\overline{\mathbf{H}}^\Upsilon = [\mathbf{H}^H, -\mathbf{v}]$ are of dimension $(U{+}1)\times B$ and $B\times(U{+}1)$, respectively. Then, step (2) is rewritten as follows:

$$\mathbf{z}^{(t+1)} = \mathbf{x}^{(t)} - \tau\overline{\mathbf{H}}^\Upsilon\overline{\mathbf{H}}\mathbf{x}^{(t)}. \qquad (4)$$

C. Proximal Operator for 3-Bit CM Precoding

The proximal operator in (3) reduces to $\mathrm{prox}_g(\mathbf{z}; \tau) = \mathrm{proj}(\frac{1}{1-\tau\gamma}\mathbf{z})$, where $\mathrm{proj}(\cdot)$ projects each element of the argument to the closest point in the polytope $\mathcal{B}$. For 3-bit CM precoding, the polytope is a regular octagon. Projecting a scalar $z \in \mathbb{C}$ onto an octagon is nontrivial so we focus on the first quadrant of the complex plane (see Fig. 2). If z is inside the octagon (in region A), then it remains there; if z is in the regions B, C, or D, then it will be mapped to j, $\frac{1}{\sqrt{2}}(1 + j)$, or 1, respectively; if z is in the regions E or F, then it will be mapped to the closest point on the lines ℓ_1 or ℓ_2, respectively. To determine in which of the six regions A–F the argument is located, we use the equations for the lines that separate them:

$$\ell_1: \ \Im(z) = (1 - \sqrt{2})\Re(z) + 1,$$

$$\ell_3: \ \Im(z) = \tfrac{1}{\sqrt{2}-1}\Re(z) + 1, \quad \ell_4: \ \Im(z) = \tfrac{1}{\sqrt{2}-1}\Re(z) - 1.$$

The equations for the lines ℓ_2, ℓ_5, and ℓ_6 are identical to the ones of ℓ_1, ℓ_4, and ℓ_3, but with $\Im(z)$ and $\Re(z)$ exchanged. Using these equations, we can project z onto the set $\mathcal{B}$.

IV. VLSI Architecture and Implementation Results

A. Architecture Overview

The proposed VLSI architecture is shown in Fig. 3 and builds upon the one of C2PO in [1], which was designed for 2-bit CM precoding. As in [1], we assume that B is a multiple of U, so the architecture consists of B/U linear arrays, each containing $U + 1$ processing elements (PEs). Each linear array operates on
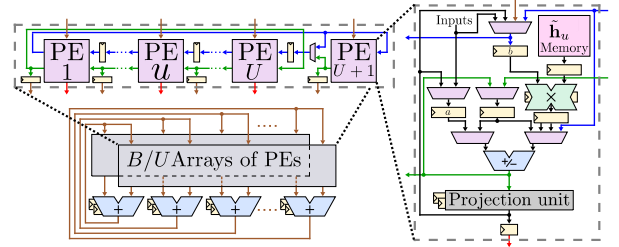

Fig. 3. High-level block diagram of the VLSI architecture for C3PO. We use B/U linear arrays, each consisting of $U + 1$ processing elements (PEs).

a $(U + 1) \times U$ sub-matrix of $\overline{\mathbf{H}}$ and on a U-dimensional sub-vector of $\mathbf{x}^{(t)}$. The architecture computes step (2) simplified as in (4) via two separate matrix-vector products using Cannon's algorithm [16]. We first compute $\mathbf{w} = \overline{\mathbf{H}}(\tau\mathbf{x}^{(t)})$ by cyclically exchanging the entries of $\tau\mathbf{x}^{(t)}$ between the PEs of the same array. We then compute $\mathbf{z}^{(t+1)} = \mathbf{x}^{(t)} - \overline{\mathbf{H}}^\Upsilon\mathbf{w}$ by cyclically exchanging the accumulated results of the PEs within the same array. Finally, the vector $\mathbf{z}^{(t+1)}$ is fed to a projection unit implementing step (3), thus completing one C3PO iteration. The proposed architecture requires $2U + \log_2(B/U) + 9$ clock cycles for one C3PO iteration. See [1] for more architecture details.

Each PE is equipped with (i) an $\tilde{\mathbf{h}}_u$ memory storing the uth row of the corresponding sub-matrix taken from $\overline{\mathbf{H}}$; (ii) a complex-valued multiply-accumulate (MAC) unit; and (iii) a projection unit. See [1] for details on (i) and (ii); part (iii), the projection unit, is more complicated than that of C2PO. Specifically, this unit maps the entries of $\mathbf{z}^{(t+1)}$ to the first quadrant of the complex plane and perform comparisons based on the line equations ℓ_1–ℓ_6 (see Sec. III-C) in order to perform the projection of $\mathbf{z}^{(t+1)}$ to $\mathcal{B}^B$.

B. Fixed-Point Parameters

The entries of $\mathbf{x}^{(t)}$ use 14-bit signed values with 8 fraction bits. The entries of $\tau\mathbf{x}^{(t)}$ use 14-bit signed values with 13 fraction bits. The entries of $\overline{\mathbf{H}}$ use 11-bit signed values with 8 fraction bits and are stored in look-up tables (LUTs) used as distributed RAM. The complex-valued MAC units use 18-bit signed values with 15 fraction bits when computing $\mathbf{w}$; 11 fraction bits are used when calculating $\mathbf{z}^{(t+1)}$. The adder tree uses 21 bits with 15 fraction bits. The projection unit represents the constants (e.g., $1 - \sqrt{2}$ and its reciprocal) using signed values with 4–5 bits, so no multipliers are used in the operations related to lines ℓ_1–ℓ_6. A total of 30 adders and subtractors are used within each projection unit; these components operate signed numbers with 7 fraction bits; the total bit-width varies between 14–15 bits, depending on the quantity.

C. Error-Rate Performance

Fig. 4(a) and Fig. 4(b) show uncoded bit-error rate (BER) as a function of the normalized transmit power $\varrho = B/N_0$ for different precoding algorithms and $U = 16$ UEs. Fig. 4(a) shows the BER for $B = 32$ BS antennas and BPSK; Fig. 4(b), for $B = 256$ BS antennas and 16-QAM. The simulation results are for $10{,}000$ Monte-Carlo trials and i.i.d. Rayleigh fading channels. Both C2PO and C3PO run with $t_{max} = 9$. For
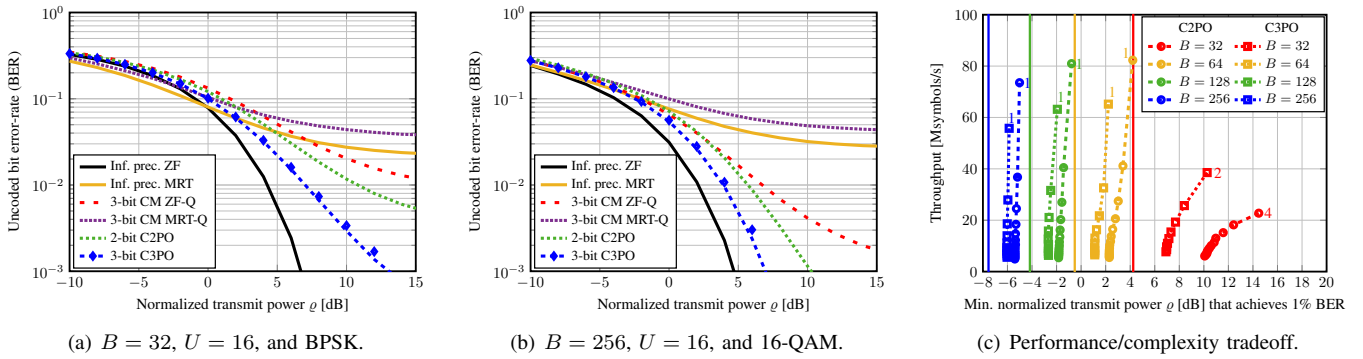
(a) $B = 32$, $U = 16$, and BPSK. (b) $B = 256$, $U = 16$, and 16-QAM. (c) Performance/complexity tradeoff.

Fig. 4. Subfigures (a) and (b): uncoded bit error-rate (BER) of various precoders versus normalized transmit power ϱ. Markers show fixed-point performance. Subfigure (c): performance/complexity tradeoffs for C2PO [1] and C3PO; the numbers next to the curves indicate $t_{\max}$. The vertical lines show the performance of infinite-precision ZF precoding. C3PO outperforms C2PO in terms of uncoded BER with an increase in implementation complexity.

TABLE I
XILINX VIRTEX-7 XC7VX690T FPGA IMPLEMENTATION RESULTS FOR MRT-Q [1], C2PO [1], AND THE PROPOSED C3PO FOR $U = 16$ UES

Algorithm	2-bit CM MRT-Q [1]				2-bit C2PO [1]				3-bit C3PO (this work)			
BS antennas B	32	64	128	256	32	64	128	256	32	64	128	256
Slices	2543	5097	9444	17630	3375	6519	12690	24748	8765	16823	33303	65451
LUTs	7842	15617	32476	64446	10817	21920	43710	85323	29034	56799	113948	224420
Flipflops	5711	11419	21902	42764	5677	12461	26083	53409	11611	24357	49893	101026
DSP48 units	0	0	0	0	136	272	544	1088	136	272	544	1088
Clock frequency [MHz]	412	410	388	359	222	206	208	193	202	175	174	157
Latency^a [clock cycles]	18	18	18	18	39	40	41	42	42	43	44	45
Throughput^a [Msymbols/s]	366	365	345	319	91	82	81	74	77	65	63	56
Power consumption^b [W]	0.79	1.25	1.84	3.16	1.04	1.70	3.17	5.80	1.76	2.89	5.48	10.12

^{a}The minimum latency and maximum throughput is measured for one algorithm iteration.
^{b}Statistical power estimation at maximum clock frequency and $1.0\,\mathrm{V}$ supply voltage.

reference, we show the BERs with 3-bit CM MRT-quantized (MRT-Q) and ZF-quantized (ZF-Q) precoding, as well as the BERs with MRT ("Inf. prec. MRT") and ZF precoding ("Inf. prec. ZF") with infinite-precision DACs. We see from Fig. 4(a) and Fig. 4(b) that the nonlinear precoders (C2PO and C3PO) significantly outperform MRT-Q and ZF-Q at high normalized transmit power ϱ. Furthermore, compared to C2PO, we note that C3PO enables a 3.75 dB gain (in terms of ϱ) at 1% uncoded BER for $B = 32$ and BPSK, and 1.75 dB for $B = 256$ and 16-QAM. Finally, we note that the implementation loss of our hardware designs (shown with blue markers) is negligible, i.e., less than 0.15 dB at 1% uncoded BER.

D. FPGA Implementation Results and Comparison

Table I shows FPGA implementation results for 2-bit CM MRT-Q [1], C2PO [1], and C3PO. All designs were developed using Verilog, and implemented using Xilinx Vivado Design Suite for a Xilinx Virtex-7 XC7VX690T FPGA. The designs support $U = 16$ UEs and were implemented for $B = \{32, 64, 128, 256\}$. Table I reveals that the resources of all designs increase roughly linearly with B. MRT-Q achieves the highest throughput thanks to its simplicity, which comes at the cost of a poor uncoded BER performance. C2PO uses $\sim 1.4\times$ more LUTs than MRT-Q and requires increased latency and critical path. Compared to C2PO, C3PO consumes $\sim 2.6\times$ the number of slices and LUTs, $\sim 2\times$ the number of flip-flops, and the same number of DSP48s. This difference is caused by

the 3-bit CM projection unit, which also increases the latency with its pipeline registers. However, C3PO can significantly outperform C2PO in terms of BER (cf. Fig. 4(a) and Fig. 4(b)).

E. Performance/Complexity Tradeoffs

Fig. 4(c) shows the performance-complexity tradeoffs of C2PO and C3PO: the complexity is represented by the minimum normalized transmit power ϱ that is required to achieve 1% uncoded BER for BPSK; the performance, by the throughput. The tradeoffs show systems with BPSK, $U = 16$ UEs and $B = \{32, 64, 128, 256\}$ BS antennas. As a reference, the minimum transmit power required for infinite-precision ZF precoding to achieve 1% uncoded BER is shown as a vertical line. We see from Fig. 4(c) that, while C2PO is able to achieve higher throughput than C3PO, C3PO requires lower transmit power to achieve 1% uncoded BER. This difference increases for small array sizes: for a system with $B = 32$, 4 iterations of C3PO achieve 1% uncoded BER at $\varrho = 8$ dB while C2PO is unable to achieve 1% uncoded BER at such value of ϱ.

V. CONCLUSIONS

We have proposed a nonlinear precoder for 8-phase (3-bit) CM transmission, C3PO, which builds upon the 4-phase C2PO precoder [1]. By using a different projection unit and no more than $2.7\times$ higher FPGA resources, C3PO achieves up to 3.75 dB transmit power reduction, and thus, low uncoded BERs in scenarios for which C2PO exhibits poor error-rate performance.

REFERENCES

[1] O. Castañeda, S. Jacobsson, G. Durisi, M. Coldrey, T. Goldstein, and C. Studer, "1-bit massive MU-MIMO precoding in VLSI," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 7, no. 4, pp. 508–522, Dec. 2017.

[2] F. Rusek, D. Persson, B. Kiong, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[3] E. G. Larsson, F. Tufvesson, O. Edfors, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[4] L. Lu, G. Ye Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, Oct. 2014.

[5] A. Mezghani, R. Ghiat, and J. A. Nossek, "Transmit processing with low resolution D/A-converters," in *Proc. IEEE Int. Conf. Electron., Circuits, Syst. (ICECS)*, Yasmine Hammamet, Tunisia, Dec. 2009, pp. 683–686.

[6] A. K. Saxena, I. Fijalkow, and A. L. Swindlehurst, "Analysis of one-bit quantized precoding for the multiuser massive MIMO downlink," *IEEE Trans. Signal Process.*, vol. 65, no. 17, pp. 4624–4634, Sep. 2017.

[7] Y. Li, C. Tao, A. L. Swindlehurst, A. Mezghani, and L. Liu, "Downlink achievable rate analysis in massive MIMO systems with one-bit DACs," *IEEE Commun. Lett.*, vol. 21, no. 7, pp. 1669–1672, Jul. 2017.

[8] S. Jacobsson, G. Durisi, M. Coldrey, T. Goldstein, and C. Studer, "Quantized precoding for massive MU-MIMO," *IEEE Trans. Commun.*, vol. 65, no. 11, pp. 4670–4684, Nov. 2017.

[9] ——, "Nonlinear 1-bit precoding for massive MU-MIMO with higher-order modulation," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 2016, pp. 763–767.

[10] H. Jedda, J. A. Nossek, and A. Mezghani, "Minimum BER precoding in 1-bit massive MIMO systems," in *IEEE Sensor Array and Multichannel Sig. Proc. Workshop (SAM)*, Rio de Janeiro, Brazil, Jul. 2016.

[11] O. Tirkkonen and C. Studer, "Subset-codebook precoding for 1-bit massive multiuser MIMO," in *Conf. on Info. Sciences and Systems (CISS)*, Baltimore, MA, Mar. 2017.

[12] A. L. Swindlehurst, A. K. Saxena, A. Mezghani, and I. Fijalkow, "Minimum probability-of-error perturbation precoding for the one-bit massive MIMO downlink," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 6483–6487.

[13] O. Castañeda, C. Studer, and T. Goldstein, "POKEMON: A non-linear beamforming algorithm for 1-bit massive MIMO," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 3464–3468.

[14] A. Noll, H. Jedda, and J. A. Nossek, "PSK precoding in multi-user MISO systems," in *Proc. Int. ITG Workshop on Smart Antennas (WSA)*, Berlin, Germany, Mar. 2017, pp. 57–63.

[15] S. Jacobsson, O. Castañeda, C. Jeon, G. Durisi, and C. Studer, "Nonlinear phase-quantized constant-envelope precoding for massive MU-MIMO-OFDM," Oct. 2017. [Online]. Available: https://arxiv.org/abs/1710.06825

[16] L. Cannon, "A cellular computer to implement the Kalman filter algorithm," Ph.D. dissertation, Montana State University, USA, 1969.

[17] T. Goldstein, C. Studer, and R. G. Baraniuk, "A field guide to forward-backward splitting with a FASTA implementation," Nov. 2014. [Online]. Available: http://arxiv.org/abs/1411.3406

[18] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, Jan. 2009.

[19] T. Goldstein and S. Setzer, "High-order methods for basis pursuit," *UCLA CAM Report*, pp. 10–41, 2010.

[20] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, Jan. 2014.