

High-Bandwidth Spatial Equalization for mmWave Massive MU-MIMO With Processing-in-Memory

Oscar Castañeda^{ID}, Sven Jacobsson^{ID}, Giuseppe Durisi^{ID}, Tom Goldstein, and Christoph Studer^{ID}

Abstract—All-digital basestation (BS) architectures enable superior spectral efficiency compared to hybrid solutions in massive multi-user MIMO systems. However, supporting large bandwidths with all-digital architectures at mmWave frequencies is challenging as traditional baseband processing would result in excessively high power consumption and large silicon area. The recently-proposed concept of finite-alphabet equalization is able to address both of these issues by using equalization matrices that contain low-resolution entries to lower the power and complexity of high-throughput matrix-vector products in hardware. In this brief, we explore two different finite-alphabet equalization hardware implementations that tightly integrate the memory and processing elements: (i) a parallel array of multiply-accumulate (MAC) units and (ii) a bit-serial processing-in-memory (PIM) architecture. Our all-digital VLSI implementation results in 28nm CMOS show that the bit-serial PIM architecture reduces the area and power consumption up to a factor of 2× and 3×, respectively, when compared to a parallel MAC array that operates at the same throughput.

Index Terms—Millimeter wave (mmWave), massive multi-user MIMO, spatial equalization, quantization, digital ASIC design, processing-in-memory (PIM).

I. INTRODUCTION

FUTURE wireless systems are expected to rely on millimeter wave (mmWave) communication [1] that provides

extreme bandwidths, and massive multi-user multiple-input multiple-output (MU-MIMO) [2] that compensates for the path loss at mmWave frequencies and enables communication with multiple user equipments (UEs) in the same time-frequency resource. However, the combination of high-bandwidth mmWave communication with hundreds of basestation (BS) antenna elements inevitably results in excessively high baseband complexity and power consumption. In order to develop power-efficient BS designs for such systems, significant attention has been given to hybrid analog-digital architectures [3]–[7]. However, such hybrid solutions are limited in the number of transmission paths they can resolve simultaneously [7]–[9], thus degrading spectral efficiency. All-digital BS architectures are a promising alternative to mitigate this drawback [10]–[12]. While it is widely believed that all-digital BS architectures consume more power than hybrid solutions, recent results in [9], [11] indicate that the power consumption of radio-frequency (RF) and data converters in all-digital architectures is comparable to that of hybrid systems when reducing the data-converter resolution. Despite these recent findings, the power consumption and system costs of baseband processing for all-digital mmWave massive MU-MIMO architectures are largely unexplored.

A. All-Digital Spatial Equalization

In the uplink, U UEs transmit information to a BS equipped with B antennas. To recover the transmitted signals, the BS must perform spatial equalization for each received sample. For linear equalization methods, one has to compute complex-valued matrix-vector products at the rate of the baseband analog-to-digital converters (ADCs). In mmWave massive MU-MIMO systems, even such straightforward matrix-vector products will result in power-hungry digital circuitry as we are dealing with extremely high sampling rates and large equalization matrices. For example, a conventional digital circuit that computes matrix-vector products for $B = 256$ antennas and $U = 16$ UEs at a rate of 2 G vectors/s consumes 28 W and occupies 129 mm² in 28 nm CMOS [13]. The power and area will further increase when considering systems with more BS antennas, more UEs, and higher sampling rates. Consequently, all-digital BS architectures require efficient spatial equalization circuitry that minimizes power and area without degrading spectral efficiency.

The hardware complexity (in terms of power and area) of matrix-vector products can be reduced by decreasing the number of bits used to represent its operands. Previous work has focused extensively on reducing the received vector's precision, which corresponds to the use of low-resolution

Manuscript received January 31, 2020; accepted March 7, 2020. Date of publication March 30, 2020; date of current version May 6, 2020. The work of Oscar Castañeda was supported in part by ComSenTer, a Semiconductor Research Corporation (SRC) Program, in part by SRC nCORE task 2758.004, and in part by the Qualcomm Innovation Fellowship. The work of Sven Jacobsson and Giuseppe Durisi was supported in part by the Swedish Foundation for Strategic Research under Grant SM13-0028 and Grant ID14-0022, and in part by the Swedish Governmental Agency for Innovation Systems (VINNOVA) within the VINN Excellence Center Chase. The work of Tom Goldstein was supported in part by the U.S. National Science Foundation (NSF) under Grant CCF-1535902, and in part by the U.S. Office of Naval Research under Grant N00014-15-1-2676. The work of Christoph Studer was supported in part by Xilinx Inc., and in part by U.S. NSF under Grant CCF-1652065, Grant CNS-1717559, and Grant ECCS-1824379. This brief was recommended by Associate Editors Yajun Ha and Edoardo Bonizzoni. (Corresponding author: Oscar Castañeda.)

Oscar Castañeda and Christoph Studer are with the School of Electrical and Computer Engineering, Cornell Tech, New York, NY 10044 USA (e-mail: oc66@cornell.edu; studer@cornell.edu).

Sven Jacobsson is with Ericsson Research, 41756 Gothenburg, Sweden, and also with the Department of Electrical Engineering, Chalmers University of Technology, 41296 Gothenburg, Sweden (e-mail: sven.jacobsson@ericsson.com).

Giuseppe Durisi is with the Department of Electrical Engineering, Chalmers University of Technology, 41296 Gothenburg, Sweden (e-mail: durisi@chalmers.se).

Tom Goldstein is with the Department of Computer Science, University of Maryland, College Park, MD 20742 USA (e-mail: tomg@cs.umd.edu).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSII.2020.2983999

ADCs (e.g., 1 to 8 bits) at the BS of massive MU-MIMO systems [7], [9]–[11], [14], [15]. The spatial equalization matrices, however, are typically represented using high-precision numbers (e.g., 10 to 12 bits) [16], [17]. Recently, reference [13] proposed *finite-alphabet equalization*, which represents equalization matrices with low-resolution numbers while minimizing the post-equalization mean-square error (MSE). In [13], finite-alphabet equalization was shown to reduce the equalization power and area by $3.9\times$ and $5.8\times$, respectively, when using conventional digital very-large scale integration (VLSI) designs.

B. Processing-in-Memory (PIM)

While the performance of digital VLSI designs continuously increased over many decades, memory access times have not improved at the same pace. This disparity led to a so-called “memory-wall” [18] in which communication with memories causes a major bottleneck in terms of throughput and energy efficiency. *Processing-in-memory* (PIM) is an emerging compute paradigm that aims at avoiding the memory wall by co-locating logic close to memories, with the goal of minimizing time and energy required for data movement [19]. With the looming end of Moore’s Law, PIM has caught increasing attention. Existing PIM approaches focus on incorporating logic into memory processes [20], exploiting mixed-signal techniques [21], using emerging devices [22], or relying on standard CMOS logic processes with all-digital processing [23].

C. Contributions

Traditional application-specific integrated circuit (ASIC) designs are closely related to PIM, as design-specific memory structures are placed near computation elements in order to maximize throughput and energy efficiency. However, with the recent appearance of numerous PIM architectures, it is natural to ask whether PIM is useful for next-generation wireless systems, a domain which has largely benefitted from ASIC designs in the past. To shed light on this question, we evaluate two distinct VLSI designs that implement finite-alphabet equalization. The first design corresponds to an array of all-digital multiply-accumulate (MAC) units, which represents traditional ASICs. The second design corresponds to a PIM approach, which equips the bit-cells of a memory array with XNOR functionality to enable efficient, massively-parallel low-resolution matrix-vector products. To enable a fair comparison between PIM and traditional ASIC designs, we use a specialized version of the recently-proposed, all-digital PPAC, which stands for Parallel Processor in Associative Content addressable memory (CAM) [23]. For the same finite-alphabet equalization throughput, we compare both solutions in terms of area and power consumption for a 28 nm CMOS technology.

D. Notation

Uppercase bold letters denote matrices; lowercase, column vectors. For a matrix $\mathbf{A}$, the transpose, Hermitian transpose, real and imaginary parts are $\mathbf{A}^T$, $\mathbf{A}^H$, $\Re\{\mathbf{A}\}$, and $\Im\{\mathbf{A}\}$, respectively. For a vector $\mathbf{a}$, the k th entry is a_k , the ℓ_2 -norm is

$\|\mathbf{a}\|_2$, and the entry-wise complex conjugate is $\mathbf{a}^*$. $\mathbb{E}_{\mathbf{x}}[\cdot]$ is the expectation operator with respect to the random vector $\mathbf{x}$.

II. SPATIAL EQUALIZATION

A. System Model and Spatial Equalization Basics

We consider the uplink of a narrowband¹ mmWave massive MU-MIMO system where a B -antenna BS receives signals from U single-antenna UEs. The uplink narrowband input-output relation is modeled as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}, \quad (1)$$

where $\mathbf{y} \in \mathbb{C}^B$ is the received vector at the BS, $\mathbf{H} \in \mathbb{C}^{B \times U}$ is the known uplink MIMO channel matrix, $\mathbf{s} \in \mathcal{S}^U$ is the transmit data vector, with $\mathcal{S}$ being the constellation set (e.g., 16-QAM), and $\mathbf{n} \in \mathbb{C}^B$ is i.i.d. circularly-symmetric complex Gaussian noise with variance N_0 per entry. We assume that the covariance matrix of $\mathbf{s}$ is $\mathbf{C}_s = \mathbb{E}_s[\mathbf{s}\mathbf{s}^H] = E_s \mathbf{I}_U$. Furthermore, we assume perfect channel state information at the BS.

Spatial equalization produces an estimate $\hat{\mathbf{s}} \in \mathbb{C}^U$ of the transmit data vector $\mathbf{s}$ given $\mathbf{y}$ and $\mathbf{H}$. With linear equalization, the estimate $\hat{\mathbf{s}}$ can be computed as

$$\hat{\mathbf{s}} = \mathbf{W}^H \mathbf{y}, \quad (2)$$

with the spatial equalization matrix $\mathbf{W}^H \in \mathbb{C}^{U \times B}$. Typically, $\mathbf{W}^H$ is chosen to minimize the MSE defined as

$$\mathbb{E}_{\mathbf{s}, \mathbf{n}} [\|\mathbf{W}^H \mathbf{y} - \mathbf{s}\|_2^2], \quad (3)$$

which results in the linear minimum MSE (L-MMSE) equalizer, given by [24]

$$\mathbf{W}^H = (\mathbf{H}^H \mathbf{H} + \rho \mathbf{I}_U)^{-1} \mathbf{H}^H \quad (4)$$

with $\rho = N_0/E_s$. The complex-valued entries of the L-MMSE equalizer are routinely represented with high-resolution numbers (e.g., 10 to 12 bits [16], [17]). For high-bandwidth mmWave massive MU-MIMO systems, such resolution leads to excessively large and power-hungry VLSI circuits.

B. Finite-Alphabet Equalization

To arrive at more efficient equalization circuitry, finite-alphabet equalization, put forward in [13], proposes to use *finite-alphabet equalization matrices* with the following structure:

$$\mathbf{V}^H = \text{diag}(\boldsymbol{\beta}^*) \mathbf{X}^H. \quad (5)$$

Here, $\mathbf{X}^H \in \mathcal{X}^{U \times B}$ is a low-resolution matrix whose entries come from a low-cardinality finite alphabet $\mathcal{X}$ (e.g., the “1-bit” alphabet is $\{\pm 1 \pm j\}$), and $\boldsymbol{\beta} \in \mathbb{C}^U$ contains per-UE high-resolution scaling factors. The work in [13] proposes two ways to compute a finite-alphabet matrix with the form in (5): (i) quantizing the L-MMSE matrix in (4) to a finite-alphabet, called FL-MMSE, and (ii) approximately solving the finite-alphabet minimum MSE equalization (FAME) problem using forward-backward splitting, called FAME-FBS. Figure 1, which is adopted from [25, Fig. 2(b)], clearly demonstrates that FAME-FBS enables superior error-rate

¹Our analysis is also suitable for wideband channels in combination with orthogonal frequency-division multiplexing (OFDM), where the input-output relation of each subcarrier can also be modeled as in (1).

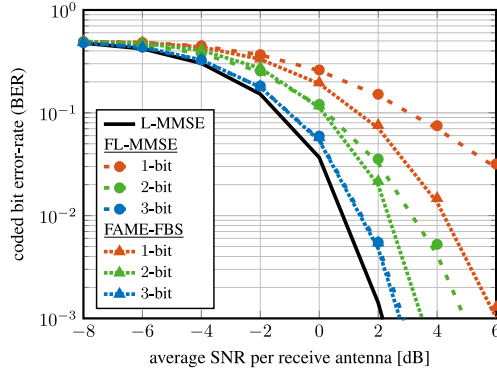


Fig. 1. Bit error-rate (BER) for a $B = 256$ BS antenna, $U = 16$ UE, 16-QAM, rate-3/4 coded OFDM mmWave MU-MIMO system operating under non-line-of-sight conditions. The curves represent the performance obtained when computing $\mathbf{W}^H \mathbf{y}$ using double-precision floating-point arithmetic; the markers represent the performance obtained when computing $\mathbf{W}^H \mathbf{y}$ using the fixed-point hardware design proposed in Section III-C. 1-bit FAME-FBS significantly outperforms 1-bit FL-MMSE. 3-bit FAME-FBS and FL-MMSE exhibit similar performance and approach the performance of the infinite-precision L-MMSE. Figure adopted from [25, Fig. 2(b)].

performance, even when considering transmission over realistic, QuaDRiGa-generated [26] mmWave non-line-of-sight channels. We note that FAME-FBS outperforms FL-MMSE as it computes finite-alphabet equalization matrices that minimize the MSE in (3).

Regardless of how the finite-alphabet matrices are computed, the structure in (5) enables efficient hardware implementations as per-UE equalization becomes

$$\hat{\mathbf{s}}_u = \mathbf{v}_u^H \mathbf{y} = \beta_u^* (\mathbf{x}_u^H \mathbf{y}), \quad (6)$$

where $\mathbf{v}_u^H$ and $\mathbf{x}_u^H$ are the u th rows of $\mathbf{V}^H$ and $\mathbf{X}^H$, respectively. Since $\mathbf{x}_u^H$ has low-precision entries, the inner product $\mathbf{x}_u^H \mathbf{y}$ (requiring B complex-valued scalar multiplications) can be computed efficiently using low-precision circuitry (e.g., adders and subtractors for the 1-bit case). The low-resolution inner product is then scaled by β_u^* , a scalar operation that is carried out at higher resolution, but only once per UE.

III. VLSI ARCHITECTURES

We will now detail three different VLSI architectures that implement finite-alphabet equalization. All of these architectures tightly integrate the datapath and memory to achieve high throughput and energy efficiency. The first and second architectures achieve such integration with a traditional ASIC design approach—the third one relies on all-digital PIM.

We consider VLSI architectures that perform matrix-vector multiplications $\mathbf{V}^H \mathbf{y}$ with a finite-alphabet equalization matrix $\mathbf{V}^H$. Furthermore, we assume that the low-resolution part $\mathbf{X}^H$ of such matrix $\mathbf{V}^H$ is represented with a symmetric set of mid-rise quantized numbers, e.g., the 2-bit alphabet $\mathcal{X}$ has entries whose real and imaginary parts are in the set $\{\pm 1, \pm 3\}$. We represent the entries of $\mathbf{y}$ using two's complement numbers.

A. Linear Array of MAC Units

As a baseline, we consider the architecture in [13], which corresponds to a linear array of U complex-valued MAC units (one per UE). Each MAC unit reads data from a memory storing the corresponding B -dimensional row $\mathbf{x}_u^H$ of $\mathbf{X}^H$ and

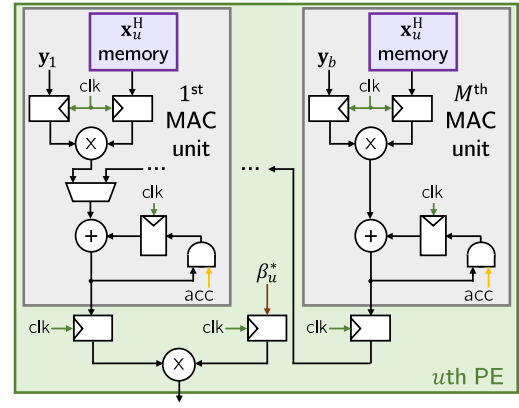


Fig. 2. Optimized MAC array architecture. A processing element (PE) is used per UE and contains M MAC units. Each of the M MAC units operates on a different section of $\mathbf{x}_u^H$, which contains B/M complex-valued numbers.

TABLE I
IMPLEMENTATION RESULTS IN 28 NM CMOS FOR ONE PPAC EQUALIZER INSTANCE OPERATING IN A SYSTEM WITH $B = 256$, $U = 16$, AND $L = 7$

$\mathbf{W}^H$ resolution K [bit]	1	2	3
Silicon area [mm ²]	0.164	0.324	0.483
Clock frequency [MHz]	796	785	784
Throughput [M vectors/s]	113.7	112.1	112.0
Power consumption ^a [mW]	112	246	383

^aExtracted from stimuli-based post-layout simulations in the typical-typical process corner at 25°C with a nominal power supply of 0.9V.

all MAC units receive one entry of $\mathbf{y}$ per clock cycle. Thus, the inner product $\mathbf{x}_u^H \mathbf{y}$ is computed in B clock cycles. The result is then scaled by β_u^* using a high-resolution multiplier.

B. Optimized MAC Array

The hardware efficiency of the baseline array of MAC units in [13] can be optimized by means of replication. As shown in Figure 2, we propose to use, for each UE, one processing element (PE) that consists of M MAC units, each MAC unit having access to a B/M -dimensional partition of $\mathbf{x}_u^H$. Then, each MAC unit within a PE receives different entries of $\mathbf{y}$ to compute the inner-product between its partitions of $\mathbf{x}_u^H$ and $\mathbf{y}$ in B/M clock cycles. The M results are then merged together to complete the inner product $\mathbf{x}_u^H \mathbf{y}$. This final reduction is achieved by reusing the adders in the MAC units following a binary-tree structure, which takes $\log_2 M$ clock cycles.

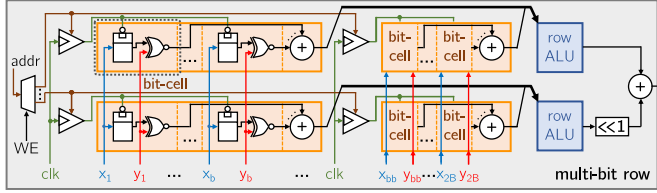
C. PPAC: Parallel Processor in Associative CAM

The PPAC architecture proposed in [23] is an all-digital CMOS PIM implementation in which every bit-cell of a memory is capable of multiplying its stored 1-bit value with an external 1-bit input using a bipolar (XNOR) or unipolar (AND) product. All products in a row are summed together by a population count in an arithmetic logic unit (ALU) associated with each row. Each row ALU can further process the population count to accelerate a range of operations, including the execution of a 1-bit matrix-vector product in a single clock cycle. In this brief, we simplify the PPAC architecture to the one illustrated in Figure 3, so that it only supports the matrix-vector products of interest for spatial equalization; see [23] for more details on PPAC.

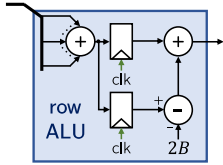
TABLE II

IMPLEMENTATION RESULTS IN 28 nm CMOS FOR VARIOUS ARCHITECTURES OPERATING AT 2 G VECTORS/S IN A SYSTEM WITH $B = 256$ AND $U = 16$

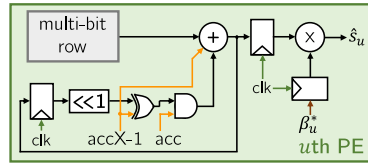
y resolution L [bits]	4									7								
Equalizer	Original MAC			Optimized MAC			PPAC			Original MAC			Optimized MAC			PPAC		
$\mathbf{W}^H$ resolution K [bit]	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Silicon area [mm ²]	21	32	42	4.1	7.0	8.6	1.8	3.6	5.3	22	33	43	5.6	9.8	11	3.0	5.8	8.7
Power consumption ^a [W]	5.0	8.3	11.8	3.8	7.2	9.1	1.2	2.7	4.2	7.2	12.0	15.9	6.4	11.1	14.2	2.0	4.4	6.9

^aExtracted from stimuli-based post-layout simulations in the typical-typical process corner at 25°C with a nominal power supply of 0.9V.

(a) Multi-bit PPAC row



(b) Row ALU details



(c) PPAC PE

Fig. 3. Parallel Processor in Associative CAM (PPAC) specialized for finite-alphabet equalization. (a) Each bit-cell includes an XNOR that carries out a 1-bit multiplication. For each matrix row, each bit-significance has its own PPAC row; the results are added with the proper scaling. (b) The row ALU sums the 1-bit products of a row and offsets them so that the result corresponds to an inner product. (c) Operation with an L -bit vector $\mathbf{y}$ is performed bit-serially across L clock cycles. One processing element (PE) is used per UE.

As in [23], our customized PPAC implementation supports multi-bit vectors in a bit-serial manner: For an input vector $\mathbf{y}$ with L -bit entries, we first input a vector $\mathbf{y}_{[L]}$ which contains the most significant bit of all $\mathbf{y}$ entries, and we compute $\mathbf{X}^H \mathbf{y}_{[L]}$, where $\mathbf{X}^H$ is stored in the PPAC memory. This process is repeated L times, while accumulating the new result to $2 \times$ the previous result (see Figure 3(c)) to compute $\mathbf{X}^H \mathbf{y}$. In contrast to [23], we implement multi-bit matrix operations by having a different row for each bit-significance of a matrix entry: If the matrix has K -bit entries, we use K PPAC rows to represent one matrix row. The results coming from each row are combined using arithmetic shifts and additions (see Figure 3(a)). Hence, each partial product $\mathbf{X}^H \mathbf{y}_{[\ell]}$, $\ell = 1, \dots, L$, is computed in a single clock cycle, completing $\mathbf{X}^H \mathbf{y}$ in L clock cycles.

While PPAC naturally operates on real-valued numbers, spatial equalization requires complex-valued operations. We use the real-valued decomposition of both $\mathbf{X}^H$ and $\mathbf{y}$, i.e., we store $\mathbf{X}_{\mathbb{R}}^T$ in PPAC and apply $\mathbf{y}_{\mathbb{R}}$ at the inputs, where

$$\mathbf{X}_{\mathbb{R}}^T = \begin{bmatrix} \Re(\mathbf{X}^H) & -\Im(\mathbf{X}^H) \\ \Im(\mathbf{X}^H) & \Re(\mathbf{X}^H) \end{bmatrix}, \quad \mathbf{y}_{\mathbb{R}} = \begin{bmatrix} \Re(\mathbf{y}) \\ \Im(\mathbf{y}) \end{bmatrix}.$$

As a result, the $U \times B$ complex-valued matrix-vector product between a matrix with K -bit entries and a vector with L -bit entries is computed in L clock cycles using a PPAC array that has $2KU$ rows, each one with $2B$ bits. The resulting U -dimensional vector $\mathbf{X}^H \mathbf{y}$ is scaled with β^* using one complex-valued multiplier per UE (see Figure 3(c)), for a

total of U multipliers. The markers in Figure 1 correspond to the fixed-point performance of our hardware design, which exhibits virtually no implementation loss when compared to double-precision floating-point arithmetic (represented by the curves).

IV. VLSI DESIGN COMPARISON

A. Comparison Methodology

We now compare VLSI implementation results for the different architectures described in Section III. We follow the procedure in [13], where, for each equalizer resolution K , a single instance of each VLSI architecture is placed-and-routed in 28 nm CMOS. Power is measured only for matrix-vector products, not for initializing the memories. Table I provides implementation results for PPAC. Since the throughput offered by a single PPAC instance is not sufficient to reach the throughputs targeted by future mmWave systems, we use several parallel instances in a time-interleaved manner to reach a target throughput of 2 G vectors/s. For example, for the case described in Table I, we need 18 PPAC instances when operating with a 1-bit equalizer. Then, we scale the area and power numbers proportionally to the number of instances.

The results for the linear array of MAC units are taken directly from [13], while the ones for PPAC are obtained from our own VLSI designs. The results for the optimized MAC array are estimated from the linear array of MAC units in the following way: For a design with K -bit equalizer resolution, let A_{MAC} be the fraction of silicon area occupied by the MAC units, excluding the $\mathbf{X}^H$ -memories, in the implementation results of the linear array of MAC units from [13]. Then, $1 - A_{\text{MAC}}$ corresponds to the fraction of the area occupied by the rest of the design, including the $\mathbf{X}^H$ -memories and β -scaling multipliers. As a result, the area occupied by the optimized MAC array with M MAC units per PE can be modeled as $A = ((1 - A_{\text{MAC}}) + M \times A_{\text{MAC}})$ times the area of the original array. We follow the same procedure to estimate the power consumption of the optimized MAC array. To compute the throughput of the optimized MAC array, we take the clock frequency of the original array from [13], and model the latency of the optimized array with $T = B/M + \log_2(M)$ clock cycles, as described in Section III-B. We choose the parameter M that minimizes the AT -product while being a power of two to perform reduction as described for the optimized MAC array.

B. Comparison of Finite-Alphabet Equalizers

Table II shows implementation results for a system with $B = 256$ antennas, $U = 16$ UEs, and for a target throughput of 2 G vectors/s for the cases where the equalizer $\mathbf{W}^H$ resolution is $K \in \{1, 2, 3\}$ bits and the received vector $\mathbf{y}$ resolution is

$L \in \{4, 7\}$ bits. For the optimized MAC array, we use $M = 16$ MAC units per UE. We observe that the optimized MAC array improves the circuit area of the original array by more than $3\times$; this is because the MAC units occupy no more than 20% of the original design's area, so that replication does not increase the per-instance area significantly, but has a significant impact on the per-instance throughput, requiring fewer instances to meet the target throughput. While there are fewer instances, each instance contains more MAC units, which draw up to 75% of the initial design's power. As a result, both MAC arrays consume roughly the same amount of power.

By comparing the optimized MAC array results to those of PPAC, we observe that PPAC achieves a significant reduction in silicon area and power consumption for all the considered cases. For example, when considering 1-bit finite-alphabet equalizers with 4-bit inputs, we see that PPAC reduces the area and power consumption by a factor of $2.3\times$ and $3.1\times$, respectively. However, these area and power consumption savings decrease as the equalizer resolution increases: When using a 3-bit equalizer with 4-bit inputs, PPAC offers $1.6\times$ and $2.2\times$ lower area and power consumption, respectively. These results illustrate that bit-serial PIM architectures provide benefits not only in scenarios that need a massive amount of memory, but also in applications that require hardware implementations with low area and power, where ASICs have been used traditionally. We finally emphasize that the theoretical advantages of PIM can already be realized with an all-digital architecture implemented in standard CMOS technology, as demonstrated by our PPAC results.

V. CONCLUSION

We have implemented finite-alphabet equalization for mmWave massive MU-MIMO systems using traditional ASIC architectures and a PIM architecture. Our implementation results have shown that an all-digital PIM solution reduces the silicon area and power consumption by at least a factor of $1.2\times$ and $2.1\times$, respectively, compared to a conventional ASIC. Combined with the savings provided by finite-alphabet equalization (up to $5.8\times$ and $3.9\times$ reduction in area and power consumption, respectively), PIM architectures pave the way for high-throughput and low-power all-digital BS architectures.

There are many avenues for future work. A system-level analysis that studies the effects of the different input interfaces used by the considered architectures is in order. Furthermore, other PIM designs that use, e.g., mixed-signal techniques or that rely on emerging devices, should be considered, as they could achieve superior throughput, area, and power consumption.

REFERENCES

- [1] A. L. Swindlehurst, E. Ayanoglu, P. Heydari, and F. Capolino, "Millimeter-wave massive MIMO: The next wireless revolution?" *IEEE Commun. Mag.*, vol. 52, no. 9, pp. 56–62, Sep. 2014.
- [2] 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.
- [3] W. Roh *et al.*, "Millimeter-wave beamforming as an enabling technology for 5G cellular communications: Theoretical feasibility and prototype results," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 106–113, Feb. 2014.
- [4] B. Sadhu *et al.*, "A 28GHz 32-element phased-array transceiver IC with concurrent dual polarized beams and 1.4 degree beam-steering resolution for 5G communication," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2017, pp. 128–129.
- [5] J. Du, W. Xu, H. Shen, X. Dong, and C. Zhao, "Hybrid precoding architecture for massive multiuser MIMO with dissipation: Sub-connected or fully connected structures?" *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5465–5479, Aug. 2018.
- [6] R. Magueta, D. Castanheira, A. Silva, R. Dinis, and A. Gameiro, "Hybrid multi-user equalizer for massive MIMO millimeter-wave dynamic sub-connected architecture," *IEEE Access*, vol. 7, pp. 79017–79029, 2019.
- [7] A. Alkhateeb, J. Mo, N. González-Prelcic, and R. W. Heath, Jr., "MIMO precoding and combining solutions for millimeter-wave systems," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 122–131, Dec. 2014.
- [8] E. Björnson, L. Van der Perre, S. Buzzi, and E. G. Larsson, "Massive MIMO in sub-6 GHz and mmWave: Physical, practical, and use-case differences," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 100–108, Apr. 2019.
- [9] S. Dutta, C. N. Barati, A. Dhananjay, D. A. Ramirez, J. F. Buckwalter, and S. Rangan, "A case for digital beamforming at mmWave," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 756–770, Feb. 2020.
- [10] J. Mo and R. W. Heath, Jr., "Capacity analysis of one-bit quantized MIMO systems with transmitter channel state information," *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5498–5512, Oct. 2015.
- [11] K. Roth and J. A. Nossek, "Achievable rate and energy efficiency of hybrid and digital beamforming receivers with low resolution ADC," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2056–2068, Sep. 2017.
- [12] S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, and C. Studer, "Throughput analysis of massive MIMO uplink with low-resolution ADCs," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 4038–4051, Jun. 2017.
- [13] O. Castañeda, S. Jacobsson, G. Durisi, T. Goldstein, and C. Studer, "Finite-alphabet MMSE equalization for all-digital massive MU-MIMO mmWave communication," *IEEE J. Sel. Areas Commun.*, to appear.
- [14] C. Studer and G. Durisi, "Quantized massive MU-MIMO-OFDM uplink," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2387–2399, Jun. 2016.
- [15] H. Yan, S. Ramesh, T. Gallagher, C. Ling, and D. Cabric, "Performance, power, and area design trade-offs in millimeter-wave transmitter beamforming architectures," *IEEE Circuits Syst. Mag.*, vol. 19, no. 2, pp. 33–58, May 2019.
- [16] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1754–1765, Jul. 2011.
- [17] M. Wu, B. Yin, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 916–929, Oct. 2014.
- [18] W. A. Wulf and S. A. McKee, "Hitting the memory wall: Implications of the obvious," *ACM SIGARCH Comput. Architect. News*, vol. 23, no. 1, pp. 20–24, Mar. 1995.
- [19] R. Nair, "Evolution of memory architecture," *Proc. IEEE*, vol. 103, no. 8, pp. 1331–1345, Aug. 2015.
- [20] S. Li, D. Niu, K. T. Malladi, H. Zheng, B. Brennan, and Y. Xie, "DRISA: A DRAM-based reconfigurable in-situ accelerator," in *Proc. IEEE/ACM Int. Symp. Microarchitect. (MICRO)*, Oct. 2017, pp. 288–301.
- [21] H. Jia, Y. Tang, H. Valavi, J. Zhang, and N. Verma, "A microprocessor implemented in 65nm CMOS with configurable and bit-scalable accelerator for programmable in-memory computing," Nov. 2018. [Online]. Available: arXiv:1811.04047.
- [22] Q. Guo, X. Guo, R. Patel, E. Ipek, and E. G. Friedman, "AC-DIMM: Associative computing with STT-MRAM," *ACM SIGARCH Comput. Architect. News*, vol. 41, no. 3, pp. 189–200, Jun. 2013.
- [23] O. Castañeda, M. Bobbett, A. Gallyas-Sanhueza, and C. Studer, "PPAC: A versatile in-memory accelerator for matrix-vector-product-like operations," in *Proc. IEEE Int. Conf. Appl. Specific Syst. Architect. Process. (ASAP)*, Jul. 2019, pp. 149–156.
- [24] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
- [25] O. Castañeda, S. Jacobsson, G. Durisi, T. Goldstein, and C. Studer, "Soft-output finite alphabet equalization for mmWave massive MIMO," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, to appear.
- [26] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, Jun. 2014.