Hardware Topologies for Decentralized Large-Scale MIMO Detection Using Newton Method

Abhinav Kulkarni, Messaoud Ahmed Ouameur, *Member, IEEE*, and Daniel Massicotte, *Senior Member, IEEE*

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

***Abstract*— Centralized Massive Multiple Input Multiple Output (MIMO) uplink detection techniques for baseband processing possess severe bottleneck in terms of interconnect bandwidth and computational complexity. This problem has been addressed in the current work by adapting the centralized Newton method for decentralized MIMO uplink detection leveraging several Base Station antenna clusters. The proposed decentralized Newton (DN) method achieves error-rate performance close to centralized Zero Forcing detector as compared to other decentralized techniques. Two hardware topologies, namely the ring and the star topologies, are proposed to assess and discuss the trade-off among interconnect bandwidth and throughput, in comparison with contemporary decentralized MIMO uplink detection techniques. As such the following findings are elaborated. On BS antenna cluster scaling for different MIMO system configurations, the ring topology provides high throughput at constant interconnect bandwidth, while the star topology provides lower latency with a deterministic variation in the hardware resource consumption. Due to strategic optimizations on the hardware implementation, additional user equipment can be allotted at a fractional increase in Field Programmable Gate Array resource consumption, improved energy efficiency, and increased transaction of bits per Joule. The ring topology can process additional subcarrier at a fractional increase in latency and improved system throughput.**

***Index Terms*— MIMO uplink detection, Newton method, FPGA, decentralized processing, hardware topology, interconnect bandwidth.**

I. Introduction

BEING a promising concept for future cellular networks, Massive Multiple Input Multiple Output (MIMO) technology has now made its way to 5G as one of the means to substantially improve both spectral and energy efficiencies [1], [2]. Future trends for 6G suggest the use of Extremely Large Aperture Array (ELAA) to provide order-of-magnitude higher area throughput compared to what massive MIMO with compact arrays can ultimately deliver [3]. It is possible for a BS to service several UEs simultaneously within the same time-frequency resources using hundreds or thousands of BS antennas. Centralized linear processing techniques for MIMO uplink signal detection like Zero Forcing (ZF), Minimum Mean Square Error (MMSE), and Maximum Ratio Combining (MRC) estimate the UE's signals by Gram matrix inversion but have caveats on hardware implementation due to high computational complexity and impose severe bottleneck in terms of interconnect bandwidth as well [1].

A. Related Work

Digital signal processing architecture design with practical system constraints for the next generation Massive MIMO uplink detection techniques is presented in [4]. For a 16-QAM MIMO system configuration with 128 BS antennas (B) and 8 UEs (U), the system parameters for centralized techniques evaluated on a FPGA are compared hereafter. MIMO uplink detection based on Neumann Series (NS) [5] achieves a throughput of 402 Mbps and high error-rate performance for large B/U ratio, however this method scales to computational complexity of $\mathcal{O}(U^3)$ for 3 series expansion terms. Conjugate Gradient (CG) based MIMO uplink detection method [6] achieves a throughput of 13 Mbps with lower FPGA resource utilization and lower error-rate performance as compared to NS method. By efficiently implementing centralized Newton method [7], the MIMO uplink detector implementation achieves a staggering throughput of 610 Mbps. Co-ordinate Descent (CD) algorithm has been adapted for MIMO uplink detection in [8], achieving a throughput of 250 Mbps at low computational complexity of $\mathcal{O}(BU)$. To alleviate the high computational complexity of NS method, Gauss Seidel (GS) algorithm has been adapted for MIMO uplink detection in [9] and achieves a throughput of 32 Mbps. An improved version of GS [10] method, that uses multiple parallel sub-carrier instances by hardware interleaving, achieves a throughput of 488 Mbps. An efficient implementation of MMSE detection has been presented in [11] and achieves a throughput of 205 Mbps. By using adaptive Successive Over Relaxation (A-SOR) to achieve fast convergence, the hardware implementation in [12] achieves a throughput of 135 Mbps with $\mathcal{O}(U^2)$ computational complexity. For high energy efficiency, ASIC based implementations [11], [13] are more advantageous over FPGA based implementations [5]–[10], [12].

Manuscript received February 17, 2021; revised May 7, 2021 and June 13, 2021; accepted June 28, 2021. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), in part by Prompt, in part by the Canadian Foundation for Innovation (CFI), in part by CMC Microsystems, in part by NUTAQ Innovation, and in part by the Laboratoire des signaux et systèmes intégrés and the Chaire de recherche sur les signaux et l'intelligence des systèmes haute performance (www.uqtr.ca/lssi). This article was recommended by Associate Editor G. Jovanovic Dolecek. *(Corresponding author: Daniel Massicotte.)*

The authors are with the Department of Electrical and Computer Engineering, Université du Québec à Trois-Rivières, Trois-Rivières, QC G9A 5H7, Canada (e-mail: abhinav.kulkarni@uqtr.ca; messaoud.ahmed.ouameur@uqtr.ca; daniel.massicotte@uqtr.ca).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2021.3097042.

Digital Object Identifier 10.1109/TCSI.2021.3097042

Centralized baseband processing techniques are feasible on hardware for a small number of UE and a low number of BS antennas for real-time processing. However, as the number of UE grows, more BS antennas are required to achieve optimal performance which increases interconnect bandwidth between BS antennas and BS central processing unit [3]. Also, all Channel State Information (CSI) has to be transferred from BS antennas to the BS central processing unit which increases computational complexity and latency at BS, thereby decreasing system throughput [14] and possess a bottleneck to ELAA implementation [3]. To address this bottleneck, several decentralized baseband processing algorithms and accompanying architectures for MIMO uplink detection have been proposed, where the baseband processing of MIMO uplink signal detection is shared by several BS antenna clusters. Decentralized Co-ordinate Descent (DCD) [15] based MIMO uplink detection computes partial uplink signal at every distributed BS antenna cluster using co-ordinate descent method. Partial signal estimates are scaled by BS antenna cluster variance and fused to produce the final uplink signal at the BS. Decentralized Alternating Direction Method of Multipliers (D-ADMM) [16] is a high computational complexity method based on consensus exchange, providing near MMSE performance with few iterations for low UE load with respect to BS antenna cluster. ADMM-GS [17] embeds Gauss-Siedel iteration in ADMM for performance enhancement in terms of error rate. This method is suited for high SNR and high UE load scenarios and is robust to channel estimation errors. The Decentralized Conjugate Gradient (D-CG) [16] method provides near MMSE performance with few iterations in high UE load scenarios per BS antenna cluster. MIMO uplink detection with Stochastic Gradient Descent (SGD) [18] uses fully decentralized architecture in Daisychain topology. In this technique, the central cluster does not have to be reconfigured when adding new BS antenna clusters, and the interconnect bandwidth between two clusters remains constant for a given number of UE.

Maximum A Posteriori (MAP) estimate based decentralized algorithms like large-MIMO approximate message passing (LAMA) with two architectures one for partially decentralized (LAMA-PD) and another one for fully decentralized (LAMA-FD) [19] and Expectation Propagation (EP) [20], [21] provide high error-rate performance at expense of increased algorithm computational complexity. MIMO uplink detection using LAMA-PD and LAMA-FD [19] equalization provides optimal performance given channel matrix $\mathbf{H}$ has i.i.d distribution and profiled with the variance of $1/B$, where B represents the number of BS antennas. However, LAMA is not robust for realistic channel environments [22]. MIMO uplink detection using EP [20] is a comparable algorithm to LAMA and involves explicit matrix inversion, which increases its computational complexity. MIMO uplink detection using EP with Log Likelihood Ratio (LLR) [21] provides improved performance than LAMA, especially at high SNR, but requires high interconnect bandwidth. MAP methods incur additional computing overheads to improve numerical stability for variance computation from noise statistics. Also, the partial local estimates have to be fused and processed using a soft-detector for uplink signal estimation in MAP methods, an improvement has been suggested by [23]. Tree K-ary based MIMO uplink detection architecture [24] discusses a decentralized scalable BS system, where interconnect links grow logarithmically on the addition of BS antenna clusters. Most of these techniques are used herein as benchmarks to discuss the error rate performance, throughput, interconnect bandwidth and the computational complexity.

B. Contributions

The choice of a MIMO uplink detection technique is based on MIMO system requirements [2] and it is a non-trivial task. Hence, there is a trade-off between error-rate performance, hardware computational complexity, latency, and system throughput based on the wireless propagation environment parameters [25]. With advancements in computing and RF technology, massive MIMO will gradually evolve into extremely large-scale MIMO systems where BS will function with thousands of antennas and in such scenarios, decentralized architectures would be more favorable. With such large MIMO antenna configurations, even the MAP methods with high computational complexity show diminishing benefits[25]. In the evolving communication standards towards 6G [26], factors of interconnect bandwidth and energy efficiency would also play a prime role along with throughput and latency for large MIMO systems. In the current work, the following contributions are presented:

- The adaptation of the centralized Newton method [27], [28] for decentralized processing of MIMO uplink detection is achieved by constructing novel local objective functions over decentralized BS antenna clusters (which we refer to as clusters for brevity). The proposed decentralized Newton (DN) algorithm provides close to the ZF symbol-error rate performance as compared to contemporary decentralized MIMO uplink detection techniques, specifically in low SNR regime and 3GPP radio channel environment.
- At the *system level*, novel proposition of ring and star topological architectures for VLSI hardware implementation to achieve gradient and Hessian sampling for the DN method, leveraging decentralized clusters at the *circuit level* to achieve trade-off among throughput, latency, energy efficiency and interconnection bandwidth.
- Analytical analysis of the interconnect bandwidth of the star and ring topologies with contemporary decentralized MIMO uplink detection techniques at the *system level*. The star topology provides lower interconnect bandwidth than EP, EP-LLR and ADMM-GS. The ring topology has lower interconnect bandwidth than the star topology and maintains constant interconnect bandwidth on MIMO configuration scaling.
- Analysis of computational complexity of the star and ring topologies with contemporary MIMO uplink detection techniques at the circuit level. Interestingly, the DN algorithm's computational complexity is in order in the number of UEs and avoids signal variance computation.

- At the *system level*, design space exploration for the hardware implementation of the star and ring topologies on FPGA and analysis of the effect of MIMO configuration scaling on system throughput, latency, energy efficiency and hardware resource consumption. The star topology provides low latency while the ring topology provides higher throughput. The implementation of the ring topology with additional sub-carrier requires a fractional increase in hardware resource consumption.
- Provide a comparative analysis of hardware implementation of the star and ring topologies with hardware architectures of contemporary MIMO uplink detection techniques at the *system level*. The star and ring topologies are feasible to implement on FPGA with high energy efficiency.

For notations, uppercase bold letter represents a matrix and lowercase bold letter represents a column vector. (t) denotes t^{th} iteration. L2 vector norm is represented as $||.||_2$. $\nabla_{\mathbf{x}}$ represents first degree gradient operator w.r.t to $\mathbf{x}$. $\nabla^2_{\mathbf{x}}$ represents second degree gradient operator w.r.t to $\mathbf{x}$. $\mathbb{E}$ represents expectation operator. For a matrix $\mathbf{A} \in \mathbb{C}^{B\times U}$, $[\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_3 \ \dots \ \mathbf{a}_U]$ represents $\mathbf{A}$ as set of column vectors, where $\mathbf{a}_i \in \mathbb{C}^{B\times 1}$. For a matrix $\mathbf{A}$, $\mathbf{A}^H$ represents complex conjugate transpose of $\mathbf{A}$. The operation **diag**(.) extracts major diagonal of a square matrix as a column vector. The operation **diagdiag**(.) is the inverse of **diag**(.) and constructs diagonal matrix with given column vector as a major diagonal.

The paper is organized as follows; Section I.*A* discusses related work on Massive MIMO uplink detection techniques, specifically motivating the need for decentralized processing techniques. Section I.*B* presents the novel contributions of the current work. Section II lays the foundation for the decentralized Newton-based MIMO uplink detection technique and derives topological architectures for hardware implementation. Section III provides a comparative analysis against interconnect bandwidth for contemporary decentralized MIMO uplink detection techniques. Section IV analyses the computational complexity of contemporary decentralized MIMO uplink detection techniques. Section V discusses simulation and error rate performance analysis for decentralized MIMO uplink detection techniques. Section VI provides hardware implementation for the ring and star topologies and draws detailed hardware implementation analysis for both topologies, with comparative analysis with other decentralized MIMO uplink detection techniques. Section VII ends the discussion with the conclusion and future potential of ring and star topologies.

II. Proposed Technique

For the pre-processing of the DN method, local objective function f_c for $c = 1, 2, \dots, C$ at every cluster is constructed. For a generic BS model, a BS with B antennas serving U number of UEs is considered. Without loss of generality, every UE is assumed to be equipped with a single antenna. Expression **y=Hx+n** represents MIMO uplink signal at BS, where $\mathbf{y} \in \mathbb{C}^{B\times 1}$ is the vector representing receive signal over B antennas of the BS, $\mathbf{x} \in \mathbb{C}^{U\times 1}$ being signal estimate, which is transmitted from the UEs to BS. $\mathbf{H} \in \mathbb{C}^{B\times U}$ is the wireless channel model. $\mathbf{n} \in \mathbb{C}^{B\times 1}$ is the channel noise. $\mathbf{x}$ is mapped to Q bits symbol which form 2^Q-QAM modulation. As shown in Fig. 1, B antennas, $\mathbf{H}$ and $\mathbf{y}$ are equally distributed into C clusters such that every cluster c is characterized by local antennas B_c, local channel matrix $\mathbf{H}_c \in \mathbb{C}^{B_c\times U}$ and local received signal vector $\mathbf{y}_c \in \mathbb{C}^{B_c\times 1}$. The total number of antennas for the BS is represented as $B = \sum_{c=1}^{C} B_c$. $\mathbf{H}_c = [\mathbf{h}_{1,c} \ \mathbf{h}_{2,c} \ \mathbf{h}_{3,c} \dots \mathbf{h}_{u,c}]$ where $u = 1, 2, 3 \dots U$; $c = 1, 2, 3 \dots C$. $\mathbf{H}_c$ and $\mathbf{y}_c$ are known locally only to the cluster c and are not exchanged within clusters.

Lemma 1: Given $\mathbf{H}_c$ and $\mathbf{y}_c$ for $c = 1, 2, 3 \dots C$, uplink estimate at t^{th} iteration can be computed as:

$$\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)} - (\mathbf{D})^{-1}\left(\sum_{c=1}^{C}(\mathbf{H}_c^H\mathbf{H}_c\mathbf{x}^{(t-1)} - \mathbf{H}_c^H\mathbf{y}_c)\right) \quad (1)$$

The detailed derivation of eq. (1) is postponed to the Appendix whereas Topologies 1 and 2 show the DN algorithm (c.f. Appendix) using two different hardware topologies as depicted in Fig. 1. The algorithm is terminated at iteration $t = T$ to obtain $\mathbf{x}^{(T)}$, which is processed using QAM decoder to obtain the uplink signal estimate. While computing eq. (1) it is important to note quantities that are static for a specific interval. In MIMO uplink signal transmission, the channel statistical characteristics remain constant during a specified interval of time. This time interval is called coherent time and $\mathbf{H}_c$ for $c = 1, 2, 3 \dots C$ remains constant during the coherent time interval. Hence, the Gram matrix $\mathbf{H}_c^H\mathbf{H}_c$ for each cluster and the approximate Hessian diagonal matrix $\mathbf{D}$ at apex cluster C have to be computed once every coherent time interval. Thus matrix multiplication of $\mathbf{D}^{-1}$ with eq. (5) involves U complex divisions, which is insignificant as compared to the total complex multiplications involved in overall algorithm.

On a single cluster, eq. (10) is implemented to obtain an uplink signal estimate. However, it is essential to design architectures that can be implemented to accumulate local computations at a single cluster.

In essence, the ring and star topologies for the DN algorithm for the MIMO uplink detection algorithm are proposed. These topological architectures enable the provision of explicit trade-offs among system latency, throughput, interconnect bandwidth, energy efficiency and hardware resource consumption. Fig. 1 shows the implementation of both topological architectures. Partial computations in a cluster are represented in eq. (6) and (5).

The ring topology is characterized by clusters organized in daisy-chain fashion. Every cluster is exactly connected to two adjacent clusters. Except the apex cluster, all the clusters are identical in functionality. Thus, every cluster receives partial computations from prior cluster, appends its local partial computations and sends resultant computations to the next cluster in the daisy-chain. All the cluster interconnections are unidirectional. As shown in Fig. 1.a, cluster C acts as an apex cluster. The apex cluster provides partial computations and also computes eq. (10) to produce $\mathbf{x}^{(t)}$ at the t^{th} iteration. To facilitate the flow of partial computations between the interconnected clusters, the interconnect variables $\mathbf{p} \in \mathbb{C}^{U\times 1}$

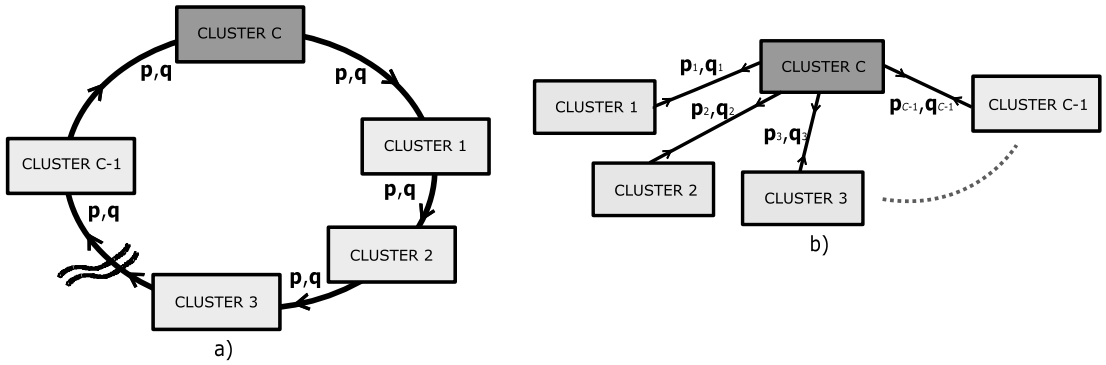


Fig. 1. DN based MIMO uplink detector implemented using a) ring topology with C clusters and interconnect variables as $\mathbf{p} \in \mathbb{C}^{U\times 1}$, $\mathbf{q} \in \mathbb{C}^{U\times 1}$ and b) star topology with C clusters and interconnect variables as $\mathbf{p}_c \in \mathbb{C}^{U\times 1}$, $\mathbf{q}_c \in \mathbb{C}^{U\times 1}$, where $c = 1, 2, 3, \ldots, C-1$. Every antenna cluster c with B_c antennas is responsible for processing local partial computations. $\mathbf{H}_c$ and $\mathbf{y}_c$ is local to every cluster and is not exchanged between clusters.

Topology 1 DN Ring Topology

Input: $\mathbf{H}_c, \mathbf{y}_c \quad c = 1, 2, 3 \ldots C$
Output: $\mathbf{x}^{(T)}$
Initialization:
Calculate $\mathbf{D}_c$ from $\mathbf{H}_c$ using eq.(8) for $c = 1, 2, 3 \ldots C$
Initial iteration $t = 1$
for $c = 1$ **to** C **do**
  $\mathbf{x}_c \leftarrow \mathbf{D}_c^{-1}(\mathbf{H}_c^H \mathbf{y}_c)$
  $\mathbf{p} \leftarrow \mathbf{p} + \mathbf{diag}(\mathbf{D}_c)$ {Accumulate: eq.(9)}
  $\mathbf{q} \leftarrow \mathbf{q} + (\mathbf{H}_c^H \mathbf{H}_c \mathbf{x}_c - \mathbf{H}_c^H \mathbf{y}_c)$ {Accumulate: eq.(5)}
  if $c = C$ **then**
    $\mathbf{D} = \mathbf{diagdiag}(\mathbf{p})$ {Local store $\mathbf{D}$ at cluster C}
    $\mathbf{x}^{(1)} \leftarrow \mathbf{x}_c - \mathbf{D}^{-1}\mathbf{q}$ {Evaluate: eq.(10)}
    $\mathbf{p} \leftarrow \mathbf{x}^{(1)}$ {Broadcast $\mathbf{x}^{(1)}$}
    $\mathbf{q} \leftarrow 0$ {Flush}
  end if
end for
for $t = 2$ **to** T **do**
  for $c = 1$ **to** C **do**
    $\mathbf{q} \leftarrow \mathbf{q} + (\mathbf{H}_c^H \mathbf{H}_c \mathbf{p} - \mathbf{H}_c^H \mathbf{y}_c)$ {Accumulate: eq.(5)}
    if $c = C$ **then**
      $\mathbf{x}^{(t)} \leftarrow \mathbf{x}^{(t-1)} - \mathbf{D}^{-1}\mathbf{q}$ {Evaluate: eq.(10)}
      $\mathbf{p} \leftarrow \mathbf{x}^{(t)}$ {Broadcast $\mathbf{x}^{(t)}$}
      $\mathbf{q} \leftarrow 0$ {Flush}
    end if
  end for
end for

Topology 2 DN Star Topology

Input: $\mathbf{H}_c, \mathbf{y}_c \quad c = 1, 2, 3 \ldots C$
Output: $\mathbf{x}^{(T)}$
Initialization:
Calculate $\mathbf{D}_c$ from $\mathbf{H}_c$ using eq.(8) for $c = 1, 2, 3 \ldots C$
Initial iteration $t = 1$
for $c = 1$ **to** C **do**
  $\mathbf{x}_c \leftarrow \mathbf{D}_c^{-1}(\mathbf{H}_c^H \mathbf{y}_c)$
  $\mathbf{p}_c \leftarrow \mathbf{diag}(\mathbf{D}_c)$
  $\mathbf{q}_c \leftarrow (\mathbf{H}_c^H \mathbf{H}_c \mathbf{x}_c - \mathbf{H}_c^H \mathbf{y}_c)$
  if $c = C$ **then**
    $\mathbf{p} = \sum_{c=1}^{C}(\mathbf{p_c})$ {Accumulate: eq.(9)}
    $\mathbf{q} = \sum_{c=1}^{C}(\mathbf{q_c})$ {Accumulate: eq.(5)}
    $\mathbf{D} = \mathbf{diagdiag}(\mathbf{p})$ {Local store $\mathbf{D}$ at cluster C}
    $\mathbf{x}^{(1)} \leftarrow \mathbf{x}_c - \mathbf{D}^{-1}\mathbf{q}$ {Evaluate: eq.(10)}
    $\mathbf{p}_c \leftarrow \mathbf{x}^{(1)}$ {Broadcast $\mathbf{x}^{(1)}$}
    $\mathbf{q}_c \leftarrow 0$ {Flush}
  end if
end for
for $t = 2$ **to** T **do**
  for $c = 1$ **to** C **do**
    $\mathbf{q}_c \leftarrow (\mathbf{H}_c^H \mathbf{H}_c \mathbf{p}_c - \mathbf{H}_c^H \mathbf{y}_c)$
    if $c = C$ **then**
      $\mathbf{q} = \sum_{c=1}^{C}(\mathbf{q_c})$ {Accumulate: eq.(5)}
      $\mathbf{x}^{(t)} \leftarrow \mathbf{x}^{(t-1)} - \mathbf{D}^{-1}\mathbf{q}$ {Evaluate: eq.(10)}
      $\mathbf{p}_c \leftarrow \mathbf{x}^{(t)}$ {Broadcast $\mathbf{x}^{(t)}$}
      $\mathbf{q}_c \leftarrow 0$ {Flush}
    end if
  end for
end for

and $\mathbf{q} \in \mathbb{C}^{U\times 1}$ are considered. For the initial iteration $t = 0$, the variable $\mathbf{p}$ accumulates diagonal vector of $\mathbf{D}_c$ from non-apex clusters to the apex cluster C. The aggregate of $\mathbf{D}_c$ for $c = 1, 2, 3 \ldots C$ is available as $\mathbf{D}$ at the apex cluster and does not need to be computed until the next coherence time interval (since $\mathbf{H}_c$ remains constant during the coherent time interval.) For the next subsequent iterations $t = 2, 3, \ldots T$, the variable $\mathbf{p}$ is set with $\mathbf{x}^{(t)}$ at the apex cluster to be broadcasted and utilized for $(t+1)^{th}$ iteration in the computation of eq. (5). Since $\mathbf{x}^{(t)}$ is available at the end of iteration t, $\mathbf{x}_c$ is the local estimate used by cluster c in the computation of eq. (5) for $t = 0$. Initial estimate $\mathbf{x}_c$ is computed from Matched Filter $\mathbf{H}_c^H \mathbf{y}_c$ and the approximate Hessian $\mathbf{D}_c$. For all the iterations, variable $\mathbf{q}$ accumulates first gradient as partial computations of eq. (5). The DN algorithm for MIMO uplink detection mapped onto the ring topology is outlined in Topology 1.

The star topology is characterized by clusters connected to a single central processing cluster. The central processing cluster is the apex cluster denoted by C. The apex cluster is connected to other $C-1$ non-apex clusters. While every non-apex cluster

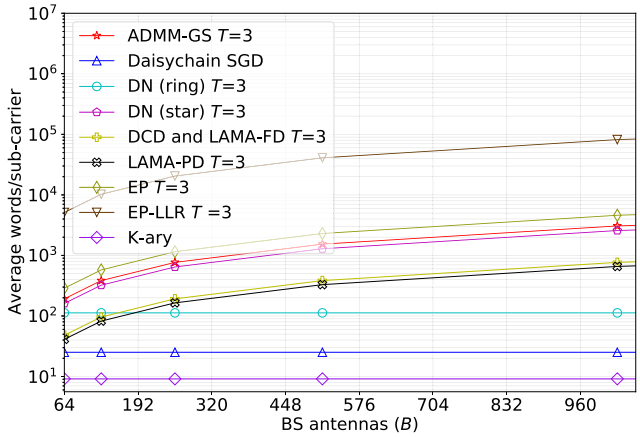


Fig. 2. Comparison of interconnect bandwidth for different decentralized MIMO uplink detection techniques with $U = 8$, $B_c = 32$ and $T = 3$.

is only connected to the apex cluster, the partial computations from all non-apex clusters are parallelly accumulated at the apex cluster. All the cluster interconnections are bidirectional. Similar to the ring topology, apex cluster provides partial computations along with eq. (10) to compute $\mathbf{x}^{(t)}$ at the t^{th} iteration. For interconnect transfers, the variables $\mathbf{p}_c \in \mathbb{C}^{U\times 1}$ and $\mathbf{q}_c \in \mathbb{C}^{U\times 1}$ for clusters $c = 1, 2, 3 \ldots C$ are considered, which handle interconnection transfer between the apex cluster and non-apex clusters. Partial computations for $\mathbf{p}_c$ and $\mathbf{q}_c$ for apex cluster $c = C$ are done internally. For initial iteration $t = 0$, the variable $\mathbf{p}_c$ accumulates diagonal vector of $\mathbf{D}_c$ from all non-apex clusters at the apex cluster to form $\mathbf{D}$, which remains constant for the coherent time interval. For the next subsequent iterations, $\mathbf{p}_c$ broadcasts $\mathbf{x}^{(t)}$ to all non-apex clusters $c = 1, 2, 3, \ldots C-1$ at the $(t+1)^{th}$ iteration. Similar to the ring topology, eq. (5) is computed using the local estimate of $\mathbf{x}_c$ for the initial iteration $t = 0$. For all the iterations, the variable $\mathbf{q}_c$ for $c = 1, 2, 3 \ldots C$ accumulates partial computations of eq. (5) to the apex cluster. The proposed DN algorithm for MIMO uplink detection mapped onto the star topology is outlined in Topology 2.

III. Interconnect Bandwidth

In centralized MIMO detection techniques, data from B antennas have to be transferred to the computing circuit of the apex cluster, which becomes a bottleneck when the detection technique is scaled to very large B as the bandwidth between antennas and computing circuit is dependent on B. So, decentralized MIMO detection techniques are employed, where B antennas are distributed into C clusters, and every cluster performs local partial computations. Local partial computations are aggregated over to the apex cluster to produce uplink signal estimation. For the apex cluster, the interconnection bandwidth is independent of B in decentralized MIMO uplink detection techniques, thereby mitigating for high data transfer between clusters. Fig.2 evaluates average interconnect transaction occurring during the coherent time interval of 1.0 millisecond (mapped to $N_{coh} = 14$ symbols) at the apex cluster.

TABLE I
Comparison of Interconnect Bandwidth

Technique	Words	Words ($B = 512$)
DCD [15]	$3CU$	384
ADMM-GS [17]	$4TCU$	1536
Daisychain SGD [18][1]	$\frac{2U^2}{N_{coh}} + 2U$	25
LAMA-PD [19]	$\frac{C\times(U^2+2N_{coh}U)}{N_{coh}}$	329
LAMA-FD [19]	$3CU$	384
EP [20]	$6TCU$	2304
EP-LLR [21][2]	$2(2T-1)UCB_c$	40960
K-ary [24][3]	$\frac{U(U+1)}{N_{coh}} + 4$	9
DN ring topology	$\frac{(2N_{coh}+1)U}{N_{coh}} + 6U(T-1)$	113
DN star topology	$\frac{(2N_{coh}+1)CU}{N_{coh}} + 4CU(T-1)$	1289

[1] Total words for formulation and filtering phase.
[2] T is the number of outer iteration.
[3] Wordlength is $2U$ for transmitting N_{coh} symbols.

For interconnect bandwidth analysis, every real entity is denoted as a word and a complex number is comprised of 2 words[14]. Interconnect bandwidth is measured by average words transferred during a coherence interval. On considering a prominent scenario for which the estimated channel in the uplink is static across a coherent time interval of N_{coh} contiguous symbols, T be the number of total iterations and C be the total number of clusters. For calculating words transacted by apex cluster for a decentralized algorithm, input and output signals are taken into account for every iteration. Interconnect transfer for LAMA-PD and LAMA-FD is given in [14]. For DN method, approximate local Hessian needs to be transmitted to the apex cluster once every coherent interval, which comprises of U words (real-valued diagonal elements). Accordingly, the apex cluster in the ring topology receives an aggregate of $(2N_{coh} + 1)U$ words during the first iteration considering all symbols in coherence interval. During subsequent iterations, the apex cluster transmits an aggregate of $4N_{coh}(I-1)U$ words and receives an aggregate of $2N_{coh}(I-1)U$ words for all symbols in coherence interval. For the star topology, the apex cluster receives an aggregate of $(2N_{coh}+1)CU$ words during the first iteration considering all symbols in coherence interval. For subsequent iterations, the apex cluster transmits an aggregate of $2N_{coh}(I-1)U$ words and receives an aggregate of $2N_{coh}(I-1)U$ words for all symbols in coherence interval. The average interconnect transfer for the ring and star topologies for coherence time interval is the average of total words transmitted and received for N_{coh} symbols for all iterations T, which is outlined in Table. I.

EP-LLR has the highest interconnect bandwidth. The interconnect bandwidth of the star topology is lower than ADMM-GS, EP and EP-LLR. For low number of BS antennas, DCD has a lower interconnect bandwidth than the ring topology. However, as the number of BS antennas increase, bandwidth of DCD also increases and surpasses that of constant bandwidth

of ring topology. DCD has a lower symbol-error rate performance with higher apex cluster computational complexity as compared to DN, which can be used to trade-off with DN even at lower number of BS antennas. LAMA-PD and LAMA-FD also have lower interconnect bandwidth than star topology, however, comparatively they have higher computational complexity and are less robust in practical wireless channel environments as investigated in [22]. The interconnect bandwidth for the star topology, EP, EP-LLR, LAMA-PD, LAMA-FD and DCD depends on the number of the clusters.

To improve upon the interconnect bandwidth performance for the DN algorithm, the ring topology exhibits lower interconnect bandwidth than LAMA-PD. The interconnect bandwidth of Daisychain SGD is lower than the ring topology, however it has a lower symbol-error rate performance by at least 3 dB and has a limitation of a single antenna per cluster. K-ary is a generic topology and has the lowest interconnect bandwidth which is provided for reference. The ring topology, Daisychain SGD and K-ary maintain constant interconnect word transfer on scaling the BS for large B antennas and the apex cluster does not have to be hardware reconfigured while varying B. Adapting the proposed DN algorithm for K-ary topology to further reduce the interconnect bandwidth is a non-trivial task and is part of ongoing research.

IV. Complexity Analysis

The computational complexity of an algorithm is mainly characterized by the number of complex multiplication and division operations. It is important to evaluate the computational complexity of MIMO uplink detection algorithms when the parameters of U, B, C, B_c, and T are varied for different MIMO system configurations. As B_c and T are fixed in scaling MIMO configuration, the critical parameters to be considered are U, B and C for analyzing the computational complexity. The MAP based MIMO uplink detection techniques of LAMA-FD, LAMA-PD, EP and EP-LLR involve exponential operation to compute the signal variance. Hence, for deriving VLSI architectures based on these algorithms, it is critical to explicitly account for the computational complexity for the implementation of exponential operation to ensure numerical stability. The performance of MAP based MIMO uplink detection algorithms depends on numerical stability, specifically at high SNR when variance becomes infinitesimally small.

The computational complexity of decentralized MIMO uplink detection algorithms is evaluated in Table. II. The order of computational complexity for the proposed DN algorithm is not affected by the choice of the topology. The EP algorithm has the highest computational complexity of third order at the non-apex clusters due to explicit matrix inversion. The LAMA-FD, LAMA-PD and ADMM-GS exhibit computational complexity of second order, however ADMM-GS does not involve the computation of the exponential operations. The Daisychain SGD exhibits second order computational complexity in terms of U and depends on the number of BS antennas B, uniform across all clusters. The proposed DN, DCD and EP-LLR exhibit linear computational complexity across the apex cluster. The DN algorithm's computational complexity is dominantly affected by computation of Gram matrix $\mathbf{H}_c^H\mathbf{H}_c$ at the non-apex clusters. However, as compared to DCD and EP-LLR, the computational complexity of the DN algorithm is lower at apex cluster.

TABLE II
Complexity Comparison

Technique	Non-apex cluster	Apex cluster	#Exponential
DCD [15]	$\mathcal{O}(U)$	$\mathcal{O}(CU)$	-
ADMM-GS [17]	$\mathcal{O}(U^2)$	$\mathcal{O}(CU)$	-
Daisychain SGD [18]	$\mathcal{O}(U^2B)$	$\mathcal{O}(U^2B)$	-
LAMA-PD [19]	$\mathcal{O}(U^2)$	$\mathcal{O}(U^2)$	$\mathcal{O}(2^Q)$
LAMA-FD [19]	$\mathcal{O}(U^2)$	$\mathcal{O}(U^2)$	$\mathcal{O}(C\times 2^Q)$
EP [20]	$\mathcal{O}(U^3)$	$\mathcal{O}(CU)$	$\mathcal{O}(2^Q)$
EP-LLR [21]	$\mathcal{O}(U)$	$\mathcal{O}(BU)$	$\mathcal{O}(CU\times 2^Q)$
DN	$\mathcal{O}(U)$	$\mathcal{O}(U)$	-

V. Error Rate Performance Analysis

Decentralized MIMO detection techniques are compared by performing simulation of BS with 128 antennas servicing 8 UEs in Gaussian i.i.d and 3GPP SCM as channel models. Also, each of these system configurations is simulated with 16-QAM to analyze the effect of modulation scheme over symbol error-rate performance. DN method is simulated with the floating-point as well as fixed-point (inline with HLS analysis). For the simulation, 32-bit data type with 16-bit for the real part and 16-bit for the imaginary part of complex number representation is used, both for floating and fixed-point analysis. Since, the ring and the star topologies are architectures for VLSI hardware implementations, they do not affect the symbol error-rate performance of the proposed DN algorithm since both equate eq. (1) using the algorithm outlined in Appendix X. All arithmetic operations for the simulation are performed using Python Numpy [31] and Python Mpmath [32].

Fig. 3.a and 3.b compares the symbol-error rate performance of the MIMO detection techniques with i.i.d Gaussian channel model. For a realistic channel model, the statistical model of a correlated fading channel model[33] for the channel matrix $\mathbf{H}_c$ is represented by $\mathbf{H}_c = \Theta_{BS}^{1/2}\mathbf{A}_{i.i.d}\Theta_{UE}^{1/2}$, where $\mathbf{A}_{i.i.d} \in \mathbb{C}^{B\times U}$ represents i.i.d Rayleigh fading channel, while $\Theta_{BS} \in \mathbb{C}^{B\times B}$ and $\Theta_{UE} \in \mathbb{C}^{U\times U}$ are correlation matrices for BS and UE respectively. Using the correlated fading model, 3GPP SCM channel correlation matrices are generated [34], [35] for BS and UE. An urban scenario with micro cell distribution is assumed for channel matrix generation, where the users are randomly distributed within a cell radius of 500m. The carrier frequency is set to 3.5GHz while the BS antenna elements spacing is half the wave length. Fig. 3.c and 3.d compares symbol-error rate performance of MIMO detection techniques for 3GPP Spatial Channel Model.

Overall, EP algorithm provides the best symbol-error rate performance, while Daisychain SGD requires the highest SNR to converge. EP algorithm provides optimal performance at cost of high interconnect bandwidth at low SNR. Output

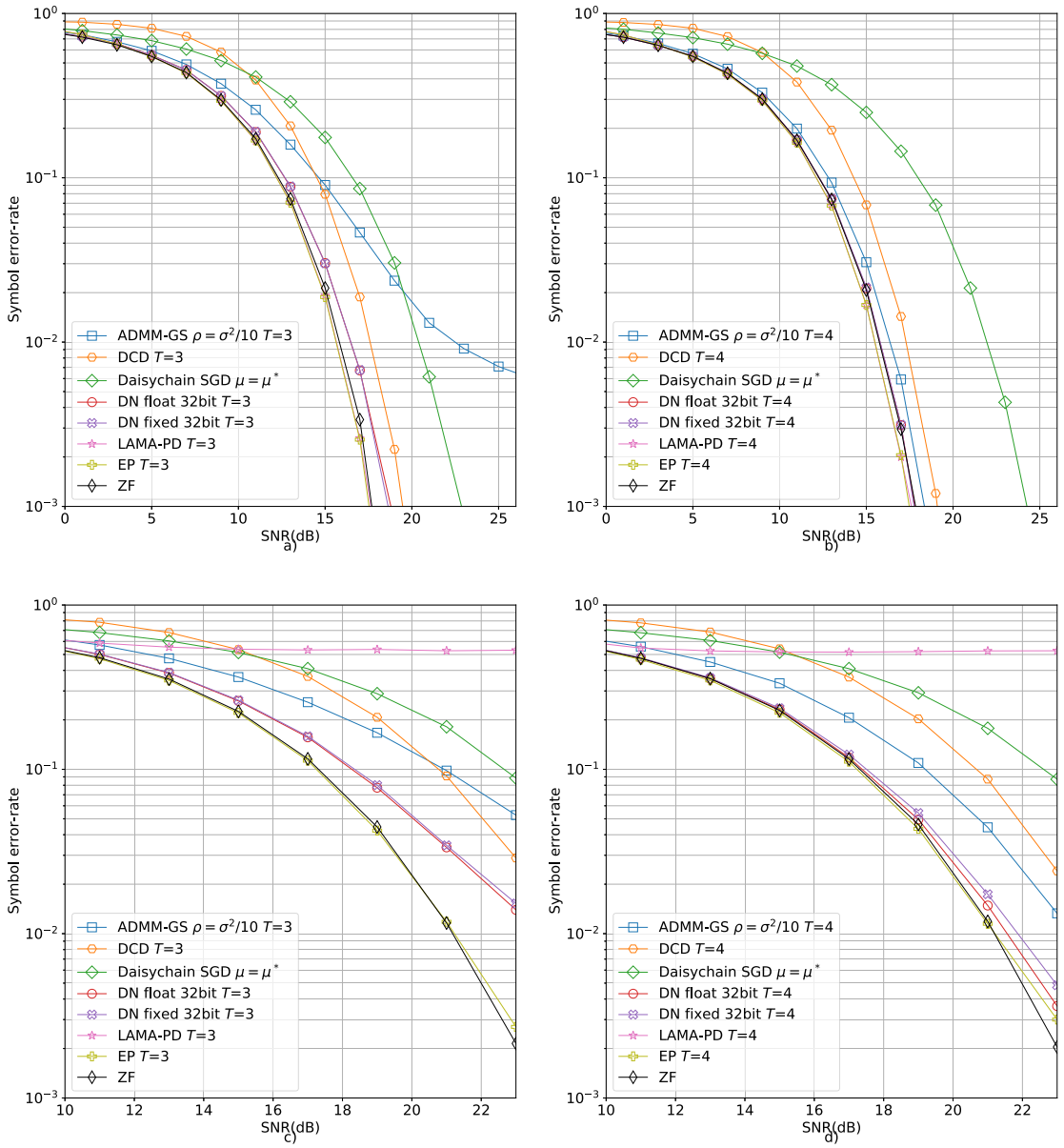


Fig. 3. Performance comparison of MIMO uplink detection algorithms for MIMO system configuration of $B = 128$, $U = 8$, $C = 4$ for 16-QAM modulation in i.i.d Gaussian channel model with 3 iterations (a) and 4 iterations (b) and in 3GPP Spatial Channel Model with 3 iterations (c) and 4 iterations (d).

equalization by EP is followed by soft-output detection [25], involving noise statistics computation. ADMM-GS is a second order algorithm which achieves fast convergence with increase in T. LAMA-PD and LAMA-FD do not converge for realistic 3GPP SCM[22]. DCD has linear computational complexity and achieves slower convergence as compared to ADMM-GS with increase in T. For the proposed DN method, the Hessian is approximated using diagonal dominance characteristics of the matrix $\mathbf{H}_c^H\mathbf{H}_c$ for $c = 1, 2, 3, \ldots C$, which saves interconnect bandwidth and provides close to ZF performance using hard-output detection.

VI. Hardware Implementation Performance Analysis

An FPGA is a reconfigurable computing technology for VLSI implementation, the design flow being different than ASIC. For the ring and star topologies, the XILINX VIRTEX-7 FPGA device is used for VLSI hardware implementation analysis. The fundamental pre-verified resource elements of an FPGA for VLSI implementation are Flip-Flops (FF), Look-up Tables (LUT), Digital Signal Processor slices (DSP48E), Block RAM of 18kB (BRAM_18K). Analysis of system parameters of throughput, resource consumption, latency, and energy efficiency for the ring and star topologies is performed on FPGA. As the ring topology with additional sub-carrier processing demands more FPGA resources, XILINX VIRTEX ULTRASCALE+ FPGA device is used for this analysis. Vivado HLS [36] is a high-level synthesis (HLS) tool used for VLSI hardware prototyping. In the implementation, HLS datatype x_complex [36] is used, which performs arithmetic bit alignment operations implicitly for complex arithmetic operations. Implementing an algorithm on FPGA and

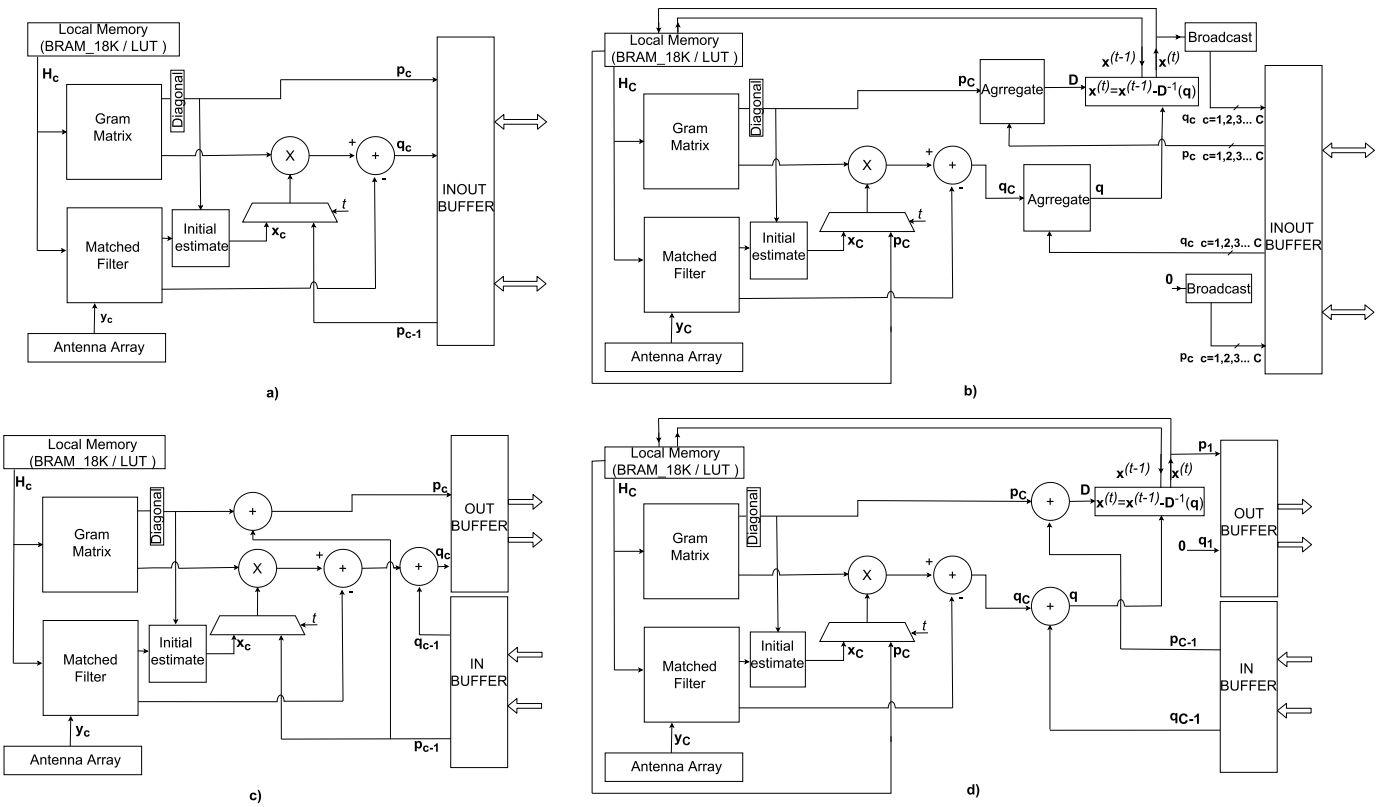


Fig. 4. The architecture diagram of MIMO uplink detection technique using the DN method. The star topology's apex cluster is shown in b) inherits the functionality of the star topology non-apex cluster shown in a). Similarly, the apex cluster for ring topology shown in d) inherits the functionality of the ring topology's non-apex cluster shown in c). All the clusters for a particular topology are implemented in a single FPGA fabric for evaluation.

TABLE III

FPGA HARDWARE RESOURCE ESTIMATES FOR DN **RING** TOPOLOGY, SINGLE SUB-CARRIER WITH 16-QAM AND $B_c = 32$ IMPLEMENTED ON XILINX VIRTEX-7 (XC7VX690T)

	Configuration	$B = 64$, $C = 2$			$B = 128$, $C = 4$		
	Number of users	$U = 2$	$U = 4$	$U = 8$	$U = 2$	$U = 4$	$U = 8$
$T = 3$	DSP48E	800 (22.2 %)	832 (23.1 %)	896 (24.9 %)	1608 (44.7 %)	1680 (46.7 %)	1824 (50.7 %)
	FF	63307 (7.3 %)	83710 (9.7 %)	122207 (14.1 %)	109849 (12.7 %)	134353 (15.5 %)	177299 (20.5 %)
	LUT	35103 (8.1 %)	51880 (12.0 %)	83162 (19.2 %)	55499 (12.8 %)	75120 (17.3 %)	111277 (25.7 %)
	BRAM_18K	4	4	8	8	8	16
	Estimated clock (MHz)	340	340	340	340	340	340
	Latency (clock cycles)	331	405	413	523	661	701
	Maximum Throughput (Mbps)	88	165	279	88	165	279
	Worst-case On-chip power (Watts)	3.573	4.095	5.080	5.342	5.992	7.177
	Power/UE (Watts)	1.786	1.024	0.635	2.671	1.498	0.897
	Mb/Joule	24.63	40.21	54.98	16.74	27.48	38.92
$T = 4$	DSP48E	1072 (29.7 %)	1120 (31.1 %)	1216 (33.7 %)	2152 (59.7 %)	2256 (62.7 %)	2464 (68.4 %)
	FF	84764 (9.8 %)	112320 (13.0 %)	164009 (18.9 %)	146832 (16.9 %)	179963 (20.8 %)	237605 (27.4 %)
	LUT	46371 (10.7 %)	68701 (15.9 %)	110398 (25.5 %)	72553 (16.7 %)	98595 (22.8 %)	146577 (33.9 %)
	BRAM_18K	4	4	8	8	8	16
	Estimated clock (MHz)	340	340	340	340	340	340
	Latency (clock cycles)	427	533	541	683	885	925
	Maximum Throughput (Mbps)	88	165	279	88	165	279
	Worst-case On-chip power (Watts)	4.302	5.004	6.325	6.643	7.519	9.108
	Power/UE (Watts)	2.151	1.251	0.790	3.321	1.880	1.138
	Mb/Joule	20.45	32.91	44.17	13.24	21.90	30.67

optimizing for performance using Vivado HLS is a non-trivial task.

HLS optimizations are applied strategically on specific arithmetic operations in the hardware architecture [37] to achieve trade-off in system latency, throughput and FPGA hardware resources utilization. HLS optimizations are applied on algorithm loop iteration or functional units. Hence, the architecture diagram for a cluster implementation of the ring and star topologies is provided in Fig. 4 for behavioral analysis, where critical arithmetic operations involving loop iterations and functional units are identified. The Gram matrix computation is the key operation with significant

computational complexity. Also, the key interconnect variables are identified between different functional units in the architecture to analyse the dependency on loop iterations.

For every cluster c, local memory registers are synthesized by BRAM_18K or LUT on the FPGA. Every cluster involves Gram Matrix ($\mathbf{H}_c^H\mathbf{H}_c$) and Matched Filter ($\mathbf{H}_c^H\mathbf{y}_c$) computations, which are computed parallelly. $\mathbf{H}_c$ is stored in local memory and utilized for every iteration. The Gram Matrix is computed at every channel coherence time interval, while Matched Filter is computed for every uplink symbol detection. Every cluster caches data from interconnect variables ($\mathbf{p}$ and $\mathbf{q}$ for the ring topology, and $\mathbf{p}_c$ and $\mathbf{q}_c$, where $c = 1, 2, 3 \ldots C$ for the star topology) using local buffers. The buffers are synthesized using BRAM_18K/LUT and operate in FIFO fashion. For the star topology, the input and the output buffers are routed to a single interconnect link, since the data flow in the interconnect variables are bidirectional. For the ring topology, the input and the output buffers are routed to separate input and output interconnect link as the data flow in interconnect variables is unidirectional.

Vivado HLS provides a pragma directive for optimizing hardware implementation [37] on FPGA to achieve a trade-off between system latency and FPGA hardware resource consumption.

A. Hardware Implementation Strategies

In Register Transfer Level (RTL) implementation, a non-apex cluster unit is built as a sub-function. For the first gradient calculation, the matrix multiplier IP core from HLS linear algebra library [36] is optimized for x_complex data-type with a fully unrolled outer row loop using pragma directive HLS UNROLL. In both topologies, every cluster (apex and non-apex) is allocated with matrix multiplier IP core with inline optimization using pragma directive HLS INLINE [37]. Inline optimization reduces processing latency of the matrix multiplier IP core at the expense of an increase in FPGA hardware resource consumption, as it constructs dedicated RTL implementation for every instance of the cluster. Dual port RAM resource implementation is used to store local channel matrix $\mathbf{H}_c$ and local receive signal $\mathbf{y}_c$. Read access to $\mathbf{H}_c^H\mathbf{H}_c$ and $\mathbf{H}_c$ are completely array partitioned in the first dimension using pragma directive HLS ARRAY_PARTITION [37]. Array partition optimization allows parallel access for every row vector of $\mathbf{H}_c^H\mathbf{H}_c$ and $\mathbf{H}_c$, which is unrolled with factor of U using pragma directive HLS UNROLL. Also, every cluster instance is pipelined using pragma directive HLS_PIPELINE [37], which reduces cluster initiation interval [36] and critical path of the cluster. In the star topology, the non-apex cluster unit only transmits $\mathbf{p}_c$ and $\mathbf{q}_c$ to the apex cluster unit. In the ring topology, every cluster unit also consists of an accumulator processing for $\mathbf{p}$ and $\mathbf{q}$ for gradient processing.

The apex cluster is built as a separate sub function and embeds functionality of non-apex cluster to calculate first gradient and the Hessian approximation. Additionally, the apex cluster performs computation of eq. (10) for every iteration t. In the star topology, the apex cluster also accumulates $\mathbf{p}_c$ and $\mathbf{q}_c$ for $c = 1, 2, 3, \ldots C$ available from the non-apex cluster, before evaluating $\mathbf{x}^{(t)}$ for t^{th} iteration. Thus, the apex cluster in the star topology has C links for each $\mathbf{p}_c$ and $\mathbf{q}_c$ where $c = 1, 2, 3, \ldots C$. Complex division for eq. (10) performed at apex cluster C for both topologies is fully unrolled using pragma directive HLS UNROLL with factor of U, which creates dedicated RTL division logic to handle each user computation of elements of $\mathbf{x}$ parallelly. All variables are implemented using dual port RAM optimized by pragma HLS RESOURCE with RAM_2P. RTL logic is realized using Configurable Logic Blocks (CLB) in FPGA [38], [39]. For the current work, the ring and star topologies are implemented on single FPGA fabric, which uses programmable interconnects between Configurable Logic Blocks (CLB) for routing algorithm.

In the ring topology, the top-level HLS synthesis function instantiates apex cluster and non-apex clusters and creates dedicated RTL implementation for every iteration of cluster instantiation. This is achieved by completely unrolling the top-level HLS synthesis function by using pragma directive HLS UNROLL [37]. Variables $\mathbf{p}$ and $\mathbf{q}$ are updated after every cluster processing for every iteration as outputs and become inputs to the next cluster in ring order. This dependence is explicitly enforced on the interconnect variables $\mathbf{p}$ and $\mathbf{q}$ using pragma HLS DEPENDENCE with Read-After-Write (RAW) option, which ensures these variables are read by the next cluster only after the write operation is performed by the current cluster. The sequential nature of ring topology enables complete unrolling of ring topology implementation to process additional sub-carrier in parallel.

In the star topology, non-apex clusters are connected to the apex cluster using dedicated variables $\mathbf{p}_c$ and $\mathbf{q}_c$ where $c = 1, 2, 3, \ldots C$. In the top-level HLS synthesis function, all non-apex clusters are instantiated with dedicated RTL logic using pragma HLS UNROLL. Every non-apex cluster c updates associated $\mathbf{p}_c$ and $\mathbf{q}_c$ parallelly and is conveyed back to apex cluster for computing $\mathbf{p}$ and $\mathbf{q}$ and thereby $\mathbf{x}^{(t)}$ for the t^{th} iteration. The assembly of the non-apex clusters and the apex cluster is unrolled for every iteration using pragma HLS UNROLL directive. Interconnect variables $\mathbf{p}_c$ and $\mathbf{q}_c$ for non-apex clusters are enforced with RAW dependence using pragma HLS DEPENDENCE for subsequent iterations.

After running the behavioral simulation for the ring and the star topologies, the respective architecture is synthesized taking account of the HLS optimizations in the RTL. After resolving critical paths, the cluster computing is time scheduled as given in Fig. 5 for MIMO configuration of $U = 8,\ B = 128,\ C = 4$ and $t = 3$. For ring topology time scheduling as shown in Fig. 5.a, owing to inter-dependency among the interconnect variables, the clusters are scheduled sequentially for every iteration $t = 1, 2, 3 \ldots T$. For the star topology, the time schedule as shown in Fig. 5.b, the non-apex clusters are scheduled parallelly since every non-apex cluster has dedicated interconnect variables with no inter-dependency. After every time interval of the non-apex cluster computation, the apex cluster computes $\mathbf{x}^{(t)}$ for every iteration t. The channel matrix $\mathbf{H}_c$ is accessed from the local memory and $\mathbf{y}_c$ is accessed from the RF frontend at initial iteration

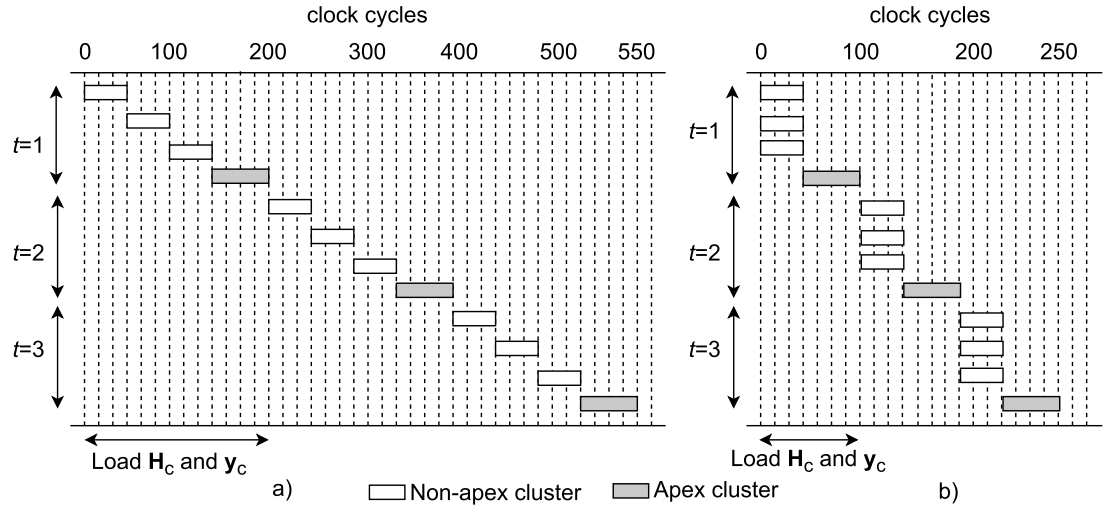


Fig. 5. The scheduling diagram from Vivado HLS Schedule viewer for the ring topology as shown in a) and the star topology as shown in b) for MIMO system configuration of $B = 128$, $U = 2$, $C = 4$, $B_c = 32$ and $T = 3$. For both topologies, $\mathbf{H}_c$ and $\mathbf{y}_c$ are accessed by each cluster during the initial iteration. The clusters are unrolled using pragma HLS UNROLL for each iteration t.

$t = 0$. After time scheduling, the maximum clock frequency for the architecture is estimated by the static timing analysis.

B. Hardware Implementation Performance Analysis

In hardware analysis, evaluation of the system parameters of latency, throughput, and on-chip power consumption for various MIMO system configurations for star and ring topologies is done. Specifically, a comparison between MIMO BS with $B = 64$ and $B = 128$ is drawn, which provides insights into change in system parameters with a change in clusters. Within particular MIMO BS, insights are provided into MIMO system configuration with UEs as $U = 2, 4, 6$ and 8 to evaluate the change in system parameters with the change in the number of UEs serviced by the system. Fig. 5 describes cluster scheduling for the star and ring topology for specific MIMO system configuration. Latency in terms of the clock cycles changes with MIMO system configuration, however, the characteristic scheduling order remains constant. For power profiling, Xilinx Power Estimator [41] is used to estimate the worst-case on-chip power consumption for different MIMO configurations presented here. For profiling, ambient temperature (25°C) and 250 Linear Feet per Minute (LFM) air supply with heat sinking as environment variables are configured. Clock toggle rate of 12.5% and enable rate of 50% is used for clock simulation. Hardware implementation for ring topology and star topology presented in Table. III and Table. IV, respectively.

1) Ring Topology Resource Analysis: Table. III gives comparative analysis for the ring topology for the system parameters. When the system with $B = 64$ ($T = 3$) is scaled from $U = 2$ configuration to support $U = 8$ configuration, overall system throughput increases by 2.2×, however throughput per UE suffers a decrease of 21%. On scaling from $B = 64$ to $B = 128$ ($T = 3$), throughput variation remain similar for system to that of $B = 64$ ($T = 3$) from $U = 2$ to $U = 8$. When star topology with $B = 64$ ($T = 3$) is scaled from $U = 2$ configuration to support $U = 8$ configuration, overall system throughput increases by 2.4×, however throughput per UE suffers a decrease of 15.4%. On scaling from $B = 64$ to $B = 128$ ($T = 3$), throughput variation remain similar to that of $B = 64$ ($T = 3$) from $U = 2$ to $U = 8$. There is no throughput variation for both topologies on increasing the number of iteration.

For the ring topology, by scaling the BS station to support additional UEs, FPGA hardware resources per UE increase fractionally. When the system with $B = 64$ ($T = 3$) is scaled from $U = 2$ configuration to support $U = 8$ configuration, there is drastic change in FPGA resource consumption per UE from $U = 2$ to $U = 8$ as 72% decrease in DSP48E, 51.7% decrease in FF, 40.7% decrease in LUT but 50% increase in BRAM is observed. On comparing FPGA resource consumption per UE for specific U, DSP48E and BRAM usage gets doubled for $B = 128$ than that for $B = 64$. However, for FF and LUT consumption per UE for specific U, a higher number of UE requires less increase in FF and LUT as compared to the lower number of UE, when the system is scaled from $B = 64$ to $B = 128$. For example, from $B = 64$ to $B = 128$, FF and LUT increases by 45.4% and 33.8% respectively for $U = 8$ as compared to 73.9% and 58% increase respectively for $U = 2$. For $B = 64$ and $B = 128$, adding an iteration with the same throughput increases DSP48E, FF, and LUT by an average of 33% with no additional BRAM requirement, for all UE cases. When system is scaled from $B = 64$ to $B = 128$ with $T = 3$ for specific number of UE, latency increase is more for higher number of UE (58% for $U = 2$ as compared to 69.7% for $U = 8$). For $B = 64$ ($T = 3$), addition of 2 UE to $U = 2$ increases latency by 22.3% as compared to addition of 4 UE to $U = 4$ with just 2% increase. Whereas for $B = 128$, addition of 2 UE to $U = 2$ and 4 UE to $U = 4$ costs 26.3% and 6% increase in latency respectively. Thus, as the number of UE increase for a particular BS, additional UE can be added at a lower increase in latency. Implementing additional iteration causes an 30% average increase in latency across $B = 64$ and $B = 128$ for all UE cases.

2) Star Topology Resource Analysis: Star topology comparative analysis is presented in Table. IV. By scaling BS

TABLE IV

FPGA HARDWARE RESOURCE ESTIMATES FOR DN **STAR** TOPOLOGY, SINGLE SUB-CARRIER WITH 16-QAM AND $B_c = 32$ IMPLEMENTED ON XILINX VIRTEX-7 (XC7VX690T)

	Configuration	$B = 64$, $C = 2$			$B = 128$, $C = 4$		
	Number of users	$U = 2$	$U = 4$	$U = 8$	$U = 2$	$U = 4$	$U = 8$
$T = 3$	DSP48E	816 (22.7 %)	864 (24.0 %)	960 (26.7 %)	1632 (45.3 %)	1728 (48.0 %)	1920 (53.3 %)
	FF	60335 (7.0 %)	70488 (8.1 %)	86581 (10.0 %)	115374 (13.3 %)	130215 (15.0 %)	150236 (17.3 %)
	LUT	27713 (6.4 %)	34072 (7.9 %)	46227 (10.7 %)	50747 (11.7 %)	60054 (13.9 %)	78607 (18.1 %)
	BRAM_18K	0	0	0	0	0	0
	Estimated clock (MHz)	340	340	340	379	379	379
	Latency (clock cycles)	242	256	304	248	262	310
	Maximum Throughput (Mbps)	70	130	237	74	138	253
	Worst-case On-chip power (Watts)	3.462	3.736	4.208	5.859	6.329	7.097
	Power/UE (Watts)	1.731	0.934	0.526	2.930	1.582	0.876
	Mb/Joule	20.22	34.62	56.24	12.63	21.81	35.61
$T = 4$	DSP48E	1088 (30.2 %)	1152 (32.0 %)	1280 (35.6 %)	2176 (60.4 %)	2304 (64.0 %)	2560 (71.1 %)
	FF	78730 (9.0 %)	92031 (10.6 %)	113023 (13.0 %)	150494 (17.4 %)	169875 (19.6 %)	195763 (22.6 %)
	LUT	35683 (8.2 %)	43801 (10.1 %)	59445 (13.7 %)	65263 (15.0 %)	77061 (17.8 %)	100733 (23.2 %)
	BRAM_18K	0	0	0	0	0	0
	Estimated clock (MHz)	340	340	340	379	379	379
	Latency (clock cycles)	313	330	398	321	338	406
	Maximum Throughput (Mbps)	70	130	237	74	138	253
	Worst-case On-chip power (Watts)	4.111	4.466	5.081	7.248	7.858	8.855
	Power/UE (Watts)	2.056	1.116	0.635	3.624	1.964	1.107
	Mb/Joule	17.03	29.11	46.64	10.21	17.56	28.57

station to support additional UE, FPGA hardware resources per UE increase fractionally. On the contrary, there is a drastic change in FPGA resource consumption per UE from $U = 2$ to $U = 8$ as a 70.6% decrease in DSP48E, a 61.54% decrease in FF, 58.20% decrease in LUT is observed. FPGA resource consumption per UE for specific U, DSP48E usage gets doubled for $B = 128$ than that for $B = 64$. However, for FF and LUT consumption per UE for specific U, system servicing a higher number of UE requires less increase in FF and LUT as compared to system servicing a lower number of UE, when two clusters are added to the system with $B = 64$. For example, from $B = 64$ to $B = 128$, FF and LUT increases by 73% and 69.15% respectively for $U = 8$ as compared to 90% and 82.8% increase respectively for $U = 2$. For $B = 64$ and $B = 128$, adding an iteration with same throughput increases DSP48E, FF and LUT by an average of 33%, 30% and 28% respectively, for $U = 2, 4$ and 8. When system is scaled from $B = 64$ to $B = 128$ with three iteration, latency increase is constant at 6 clock cycles for all UE cases. Relatively, addition of two clusters to $B = 64$ costs 2.5%, 2.34% and 2.0% increase in latency for $U = 2, 4$ and 8 respectively. Implementing additional iteration causes an average of 30.5% increase in latency across $B = 64$ and $B = 128$ for all UE cases.

3) Power Consumption Analysis: For ring topology, although total power consumption increases with the number of UE, the power consumed per UE decreases, and bits per Joule increase with the number of UE for a particular BS. For instance, from scaling $U = 2$ to $U = 8$ ($T = 3$), power consumption per UE drops by 64% for $B = 64$ and 66.4% for $B = 128$, whereas bits per Joule increase by 1.23× for $B = 64$ and 1.32× for $B = 128$, making it more power-efficient. The addition of cluster increases power consumption fractionally. By scaling from $B = 64$ to $B = 128$ ($T = 3$), power consumption per UE increases by 49.5%, 46.3% and 41.2% for $U = 2, 4$ and 8 respectively, which causes drop in bits per Joule by 33.1%, 31.6% and 29.21% for $U = 2, 4$ and 8 respectively. Addition of iteration to three iteration system increases power consumption per UE by an average of 22.4% for $B = 64$ and 25.6% for $B = 128$, whereas decreases bits per Joule by an average of 18.2% for $B = 64$ and 20.36% for $B = 128$ for all UE cases. Ring topology with a single sub-carrier can process additional sub-carrier at approximately 30% reduced latency as shown in Table. V. While processing the second sub-carrier, FPGA resource consumption for DSP48E, FF, and BRAM has almost doubled while LUT consumption increases by 53%. Throughput is increased by 1.39× for two sub-carrier as compared to single sub-carrier processing, with a 61% increase in power consumption per UE and a 53% increase in bits per Joule.

TABLE V

FPGA HARDWARE RESOURCE ESTIMATES FOR DN RING TOPOLOGY, MULTIPLE SUB-CARRIERS WITH 16-QAM, $B_c = 32$, $B = 128$, $U = 8$ AND $C = 4$ ON XILINX VIRTEX ULTRASCALE+ (XCVU13P)

Number of sub-carriers	1	2
DSP48E	1824 (14.8 %)	3840 (31.2 %)
FF	175351 (5.0 %)	367241 (10.6 %)
LUT	108709 (6.3 %)	218087 (12.6 %)
BRAM_18K	16	32
Iterations (I)	3	3
Estimated clock (MHz)	383	381
Latency (clock cycles)	707	492
Maximum Throughput (Mbps)	292	697
Worst On-chip power (Watts)	9.8	15.3
Power/UE (Watts)	1.225	1.9125
Mb/Joule	29.80	45.55

TABLE VI

COMPARISON OF **DECENTRALIZED** MASSIVE MIMO UPLINK DETECTION TECHNIQUES

Detection method	LAMA-FD [19]	LAMA-PD [19]	DCD [15]	FD-LAMA[1][40]	DN ring topology[1]	DN star topology[1]
Fabric	GPU	GPU	GPU	FPGA	FPGA	FPGA
Precision point	float[2](n.a.)	float[2](n.a.)	float (32bit)	fixed[3](32bit)	fixed (32bit)	fixed (32bit)
Parallel sub-carrier instances	-	-	-	n.a.	2	1
Modulation scheme	16-QAM	16-QAM	16-QAM	QPSK	16-QAM	16-QAM
Configuration ($B \times U$)	128×8	128×8	128×8	128×8	128×8	128×8
Iterations (I)	3	3	3	3	3	3
FF	-	-	-	76270 (2.2 %)	367241 (10.6 %)	150236 (4.3 %)
LUT	-	-	-	44420 (2.6 %)	218087 (12.6 %)	78607 (4.5 %)
DSP48E	-	-	-	1197 (9.7 %)	3840 (31.2 %)	1920 (15.6 %)
BRAM_18K	-	-	-	10	32	0
Latency (ms)	0.854	0.929	0.540	0.900	1.291	0.818
Maximum Throughput (Gbps)	1.34	1.23	1.06	0.018	0.697	0.253
Power Consumption (Watts)	1200	1200	1200	n.a.	15.3	7.097
Mb/Joule	1.12	0.94	0.88	n.a.	45.36	35.61

[1] FF, LUT, DSP48E and BRAM_18K percent estimation is normalized for XILINX VIRTEX ULTRASCALE+ (XCVU13P) FPGA device.
[2] NVIDIA cuBLAS library .
[3] Complex number representation.

For the star topology, although power consumption increases with the number of UEs, the power consumed per UE decreases, and bits per Joule increase with the number of UE for particular BS similar to the ring topology. For instance, from scaling $U = 2$ to $U = 8$ ($T = 3$), power consumption per UE drops by approximately 70% for $B = 64$ and $B = 128$, whereas bits per Joule increase by approximately $1.8\times$ for $B = 64$ and $B = 128$. The addition of clusters increases power consumption and causes a decrease in bits per Joule. By scaling from $B = 64$ to $B = 128$ for $T = 3$, power consumption per UE increases by 69%, 69% and 66.5% for $U = 2, 4$ and 8 respectively, which causes drop in bits per Joule by 37.52%, 37.00% and 36% for $U = 2, 4$ and 8 respectively. Addition of iteration to three iteration system increases power consumption per UE by an average of 19.6% for $B = 64$ and 24.6% for $B = 128$, whereas decreases bits per Joule by an average of 16.4% for $B = 64$ and 19.4% for $B = 128$.

4) Comparative Performance Analysis: On comparing the ring and star topologies, the choice of topology is dependent on the trade-off among interconnect bandwidth, throughput, energy efficiency and latency. On comparing similar MIMO configurations between the ring and star topologies, ring topology provides more throughput at expense of increased latency as compared to the star topology for $B = 64$ and $B = 128$ for all UE cases. Ring topology maintains constant interconnect bandwidth on the addition of clusters, with no RTL reconfiguration required for the apex cluster or non-apex clusters on scaling MIMO configuration for B. For similar MIMO configurations, star topology provides low latency as compared to the ring topology at the expense of reduced throughput. Also, the star topology provides a more deterministic latency increase by scaling MIMO configuration by the number of UE or clusters as compared to a ring topology. On the contrary, the ring topology processes additional sub-carrier (Table. V) at a fractional increase in latency and power consumption per UE as compared to the star topology, providing high throughput gain at the expense of twice the FPGA resource consumption.

Table. VI compares decentralized MIMO detection techniques implemented as FPGA and GPU prototypes. FD-LAMA [40] is a variant of LAMA-FD, which implements a hyperbolic tangent function for LAMA iterations, thereby increasing algorithm computational complexity. LAMA algorithm is not robust for a realistic channel environment[22]. On the application note, the star topology is favorable for a 3GPP SCM scenario that needs to be scaled with a large number of clusters at low latency at expense of high interconnect bandwidth. On the contrary, the ring topology is favorable for a 3GPP SCM scenario that requires scaling for a large number of clusters at constant interconnect bandwidth and high throughput, at expense of increased latency.

VII. CONCLUSION AND FUTURE WORK

In this work, decentralized Newton (DN) algorithm for decentralized MIMO uplink is presented, which is a novel adaptation of the centralized Newton method. Also, two novel hardware architectures for hardware implementation are proposed. Also, a comparative analysis of scaling effects on parameters of throughput, latency, FPGA resource consumption, and on-chip power consumption for both topologies is carried out. The star topology is suited for MIMO configuration scenarios that demand low-latency, while ring topology can be implemented in MIMO configuration scenarios demanding higher throughput and lower interconnect bandwidth. Interestingly, it is possible to switch between topologies using smart routing hardware to route the resource blocks allocated to enhanced mobile broadband (eMMB) services and those dedicated to Ultra-High Reliability and Low Latency (URLLC) services accordingly. In terms of scaling system for thousand of BS antennas, the ring topology maintains low and constant interconnect bandwidth as compared to star topology. Also, the ring topology can process additional sub-carrier at a fractional increase in the latency and power consumption. The star topology can be scaled for a huge number of clusters without incurring high latency. DN is a comparatively low complexity algorithm providing close to ZF performance which can be implemented feasibly on FPGA. FPGA is

inherently power efficient as compared to GPU, which makes FPGA implementation of DN algorithm power-efficient than GPU implementation of other decentralized MIMO uplink detection algorithms [15], [19].

In the future work, there is scope to implement and analyze the ring and star topologies of DN method based MIMO uplink detection using multiple FPGA, which would make the decentralized implementation more modular. Although clusters with an equal number of antennas per cluster are considered, both topologies can be implemented with non-uniform distribution of BS antennas among clusters.

APPENDIX

A. Proof of Lemma 1

$f_c(\mathbf{x})$ with $f_c : \mathbb{C}^{U\times 1} \rightarrow \mathbb{R}$ is considered as local cost function of cluster c and define it as:

$$f_c(\mathbf{x}) = ||\mathbf{H}_c\mathbf{x} - \mathbf{y}_c||_2^2 = \mathbf{x}^H\mathbf{H}_c^H\mathbf{H}_c\mathbf{x} - 2\mathbf{y}_c^H\mathbf{H}_c\mathbf{x} + \mathbf{y}_c^H\mathbf{y}_c \quad (2)$$

With ensemble of sample function $f_c(\mathbf{x})$ with respect to c, $F(\mathbf{x})$ is constructed as system objective function given as $F : \mathbb{C}^{U\times 1} \rightarrow \mathbb{R} >= 0$ for robust stochastic optimization [29], defined such that:

$$F(\mathbf{x}) = \mathbb{E}\left(f_c(\mathbf{x})\right) = \frac{1}{C}\sum_{c=1}^{C} f_c(\mathbf{x}) \quad c = 1, 2, 3 \ldots C \quad (3)$$

By adapting Newton Method [27] to evaluate $\mathbf{x}^{(t)}$, where iteration $t = 1, 2, 3 \ldots T$:

$$\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)} - \left(\nabla^2_{\mathbf{x}^{(t-1)}} F(\mathbf{x}^{(t-1)})\right)^{-1} \nabla_{\mathbf{x}^{(t-1)}} F(\mathbf{x}^{(t-1)}) \quad (4)$$

where,

$$\begin{aligned}\nabla_{\mathbf{x}^{(t-1)}} F(\mathbf{x}^{(t-1)}) &= \nabla_{\mathbf{x}^{(t-1)}} \mathbb{E}\left(f_c(\mathbf{x}^{(t-1)})\right) \\ &= \frac{1}{C}\sum_{c=1}^{C}\left(\nabla_{\mathbf{x}^{(t-1)}} f_c(\mathbf{x}^{(t-1)})\right) \\ &= \frac{2}{C}\sum_{c=1}^{C}\left(\mathbf{H}_c^H\mathbf{H}_c\mathbf{x}^{(t-1)} - \mathbf{H}_c^H\mathbf{y}_c\right) \quad (5)\end{aligned}$$

and,

$$\begin{aligned}\nabla^2_{\mathbf{x}^{(t-1)}} F(\mathbf{x}^{(t-1)}) &= \nabla^2_{\mathbf{x}^{(t-1)}} \mathbb{E}\left(f_c(\mathbf{x}^{(t-1)})\right) \\ &= \frac{1}{C}\sum_{c=1}^{C}\left(\nabla^2_{\mathbf{x}^{(t-1)}} f_c(\mathbf{x}^{(t-1)})\right) \\ &= \frac{2}{C}\sum_{c=1}^{C}\left(\mathbf{H}_c^H\mathbf{H}_c\right) \quad (6)\end{aligned}$$

$\mathbf{H}_c^H\mathbf{H}_c$ is symmetrical positive-semidefinite and is decomposed arithmetically as:

$$\mathbf{H}_c^H\mathbf{H}_c = \mathbf{D}_c + \mathbf{L}_c + \mathbf{L}_c^H \quad (7)$$

where $\mathbf{D}_c$, $\mathbf{L}_c$ and $\mathbf{L}_c^H$ are diagonal, strictly lower triangular and strictly upper triangular matrices. As $\mathbf{H}_c^H\mathbf{H}_c$ is a $U \times U$ matrix, it needs $U \times U$ dimensional interconnect between the clusters. But, as $\mathbf{H}_c^H\mathbf{H}_c$ is diagonally dominant and its column vectors being mutually orthogonal, it can be approximated as $\mathbf{H}_c^H\mathbf{H}_c \approx \mathbf{D}_c$ [25], [30]. With this approximation, column vector comprising diagonal of $\mathbf{D}_c$ can be exchanged and accumulated between clusters as $U \times 1$ dimensional column vector, thus reducing interconnect bandwidth between clusters. Accordingly, column vectors of $\mathbf{H}_c$ are used to calculate $\mathbf{D}_c$.

$$(\mathbf{D}_c)_{ij} = \begin{cases} ||\mathbf{h}_{u,c}||_2^2 & \text{when } i = j = u \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

With approximation of $\mathbf{D}_c$, second gradient is calculated as:

$$\nabla^2_{\mathbf{x}^{(t-1)}} F(\mathbf{x}^{(t-1)}) = \frac{2}{C}\sum_{c=1}^{C}\left(\mathbf{H}_c^H\mathbf{H}_c\right) \approx \frac{2}{C}\sum_{c=1}^{C}(\mathbf{D}_c) \triangleq \frac{2}{C}\mathbf{D} \quad (9)$$

While using eq. (5) and eq. (9) for evaluating eq. (4), factor $\frac{2}{C}$ gets canceled:

$$\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)} - (\mathbf{D})^{-1}\left(\sum_{c=1}^{C}(\mathbf{H}_c^H\mathbf{H}_c\mathbf{x}^{(t-1)} - \mathbf{H}_c^H\mathbf{y}_c)\right) \quad (10)$$

REFERENCES

[1] M. A. Ouameur, D. Massicotte, A. M. Akhtar, and R. Girard, "Performance evaluation and implementation complexity analysis framework for ZF based linear massive MIMO detection," *Wireless Netw.*, vol. 26, pp. 1–15, Apr. 2020.

[2] M. A. Albreem, M. Juntti, and S. Shahabuddin, "Massive MIMO detection techniques: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3109–3132, 4th Quart., 2019.

[3] E. Björnson, L. Sanguinetti, H. Wymeersch, J. Hoydis, and T. L. Marzetta, "Massive MIMO is a reality—What is next?: Five promising research directions for antenna arrays," *Digit. Signal Process.*, vol. 94, pp. 3–20, Nov. 2019.

[4] L. V. der Perre, L. Liu, and E. G. Larsson, "Efficient DSP and circuit architectures for massive MIMO: State of the art and future directions," *IEEE Trans. Signal Process.*, vol. 66, no. 18, pp. 4717–4736, Sep. 2018.

[5] 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.

[6] B. Yin, M. Wu, J. R. Cavallaro, and C. Studer, "VLSI design of large-scale soft-output MIMO detection using conjugate gradients," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Lisbon, Portugal, May 2015, pp. 1498–1501.

[7] J. Chen, Z. Zhang, H. Lu, J. Hu, and G. E. Sobelman, "An intra-iterative interference cancellation detector for large-scale MIMO communications based on convex optimization," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 11, pp. 2062–2072, Nov. 2016.

[8] M. Wu, C. Dick, J. R. Cavallaro, and C. Studer, "High-throughput data detection for massive MU-MIMO-OFDM using coordinate descent," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 12, pp. 2357–2367, Dec. 2016.

[9] Z. Wu, C. Zhang, Y. Xue, S. Xu, and X. You, "Efficient architecture for soft-output massive MIMO detection with Gauss–Seidel method," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 1886–1889.

[10] C. Zhang, Z. Wu, C. Studer, Z. Zhang, and X. You, "Efficient soft-output Gauss–Seidel data detector for massive MIMO systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, early access, Oct. 26, 2019, doi: 10.1109/TCSI.2018.2875741.

[11] G. Peng, L. Liu, S. Zhou, S. Yin, and S. Wei, "A 1.58 Gbps/W 0.40 Gbps/mm^2 ASIC implementation of MMSE detection for 128×8 64-QAM massive MIMO in 65 nm CMOS," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 5, pp. 1717–1730, May 2018.

[12] A. Yu *et al.*, "Efficient successive over relaxation detectors for massive MIMO," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 6, pp. 2128–2139, Jun. 2020.

[13] C. Jeon, O. Castaneda, and C. Studer, "A 354 Mb/s 0.37 mm^2 151 mW 32-user 256-QAM near-MAP soft-input soft-output massive MU-MIMO data detector in 28 nm CMOS," in *Proc. IEEE 45th Eur. Solid State Circuits Conf. (ESSCIRC)*, Sep. 2019, pp. 127–130.

[14] K. Li *et al.*, "Design trade-offs for decentralized baseband processing in massive MU-MIMO systems," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2019, pp. 906–912.

[15] K. Li, O. Castaneda, C. Jeon, J. R. Cavallaro, and C. Studer, "Decentralized coordinate-descent data detection and precoding for massive MU-MIMO," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.

[16] K. Li, R. R. Sharan, Y. Chen, T. Goldstein, J. R. Cavallaro, and C. Studer, "Decentralized baseband processing for massive MU-MIMO systems," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 7, no. 4, pp. 491–507, Dec. 2017.

[17] M. A. Ouameur and D. Massicotte, "Efficient distributed processing for large scale MIMO detection," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, A Coruna, Spain, Sep. 2019, pp. 1–5.

[18] J. R. Sánchez, F. Rusek, O. Edfors, M. Sarajlić, and L. Liu, "Decentralized massive MIMO processing exploring daisy-chain architecture and recursive algorithms," *IEEE Trans. Signal Process.*, vol. 68, pp. 687–700, Jan. 2020.

[19] C. Jeon, K. Li, J. R. Cavallaro, and C. Studer, "Decentralized equalization with feedforward architectures for massive MU-MIMO," *IEEE Trans. Signal Process.*, vol. 67, no. 17, pp. 4418–4432, Sep. 2019.

[20] H. Wang, A. Kosasih, C.-K. Wen, S. Jin, and W. Hardjawana, "Expectation propagation detector for extra-large scale massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2036–2051, Mar. 2020.

[21] Z. Zhang, H. Li, Y. Dong, X. Wang, and X. Dai, "Decentralized signal detection via expectation propagation algorithm for uplink massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11233–11240, Oct. 2020.

[22] C. Jeon, R. Ghods, A. Maleki, and C. Studer, "Optimal data detection in large MIMO," Nov. 2018, *arXiv:1811.01917*. [Online]. Available: http://arxiv.org/abs/1811.01917

[23] P. Seidel, S. Paul, and J. Rust, "Decentralized massive MIMO uplink signal estimation by binary multistep synthesis," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2019, pp. 1967–1971.

[24] E. Bertilsson, O. Gustafsson, and E. G. Larsson, "A scalable architecture for massive MIMO base stations using distributed processing," in *Proc. 50th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2016, pp. 864–868.

[25] S. Yang and L. Hanzo, "Fifty years of MIMO detection: The road to large-scale MIMOs," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1941–1988, 4th Quart., 2015.

[26] N. Rajatheva *et al.*, "White paper on broadband connectivity in 6G," Apr. 2020, *arXiv:2004.14247*. [Online]. Available: http://arxiv.org/abs/2004.14247

[27] J. Nocedal and S. Wright, *Numerical Optimization* (Springer Series in Operations Research and Financial Engineering). New York, NY, USA: Springer, 2006.

[28] S. P. Karimireddy, S. U. Stich, and M. Jaggi, "Global linear convergence of Newton's method without strong-convexity or Lipschitz gradients," Jun. 2018, *arXiv:1806.00413*. [Online]. Available: http://arxiv.org/abs/1806.00413

[29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[30] T. L. Marzetta and E. G. Larsson, *Fundamentals of Massive MIMO*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[31] T. E. Oliphant, *Guide to NumPy*, 2nd ed. North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 2015.

[32] F. Johansson *et al.* (Dec. 2013). *Mpmath: A Python Library for Arbitrary precision Floating-Point Arithmetic (Version 0.18)*. [Online]. Available: http://mpmath.org/

[33] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, "MIMO channel models," in *MIMO-OFDM Wireless Communications With MATLAB*. Singapore: Wiley, 2010, pp. 71–109.

[34] J. Salo *et al.*, *MATLAB Implementation of the 3GPP Spatial Channel Model*, document TR 25.996, 3GPP, Jan. 2005. [Online]. Available:http://www.tkk.fi/Units/Radio/scm/

[35] *Spatial Channel Model for Multiple Input Multiple Output (MIMO) Simulations*, document TR 25.996, 3GPP, Release 16, Jul. 2020.

[36] *Vivado Design Suite User Guide High-Level Synthesis UG902 (V2019.1)*, Xilinx, San Jose, CA, USA, Jul. 2019.

[37] *SDx Pragma Reference Guide UG1253 (V2019.1)*, Xilinx, San Jose, CA, USA, Jun. 2019.

[38] *Xilinx 7 Series FPGAs Configurable Logic Block User Guide UG474 (V1.8)*, Xilinx, San Jose, CA, USA, Sep. 2016.

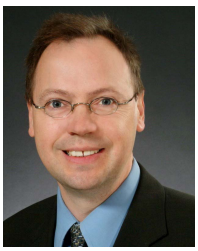
[39] *Xilinx 7 Series FPGAs Data Sheet: Overview DS180 (V2.6.1)*, Xilinx, San Jose, CA, USA, Sep. 2020.

[40] K. Li, C. Jeon, J. R. Cavallaro, and C. Studer, "Decentralized equalization for massive MU-MIMO on FPGA," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Oct. 2017, pp. 1532–1536.

[41] *Xilinx Power Estimator User Guide UG440 (V2019.2)*, Xilinx, San Jose, CA, USA, Oct. 2019.

Abhinav Kulkarni received the bachelor's degree in electronics and communication engineering from VNIT, India, in 2016, and the master's degree in embedded systems from Nanyang Technological University, Singapore, in 2017. He is currently pursuing the Ph.D. degree in electrical engineering with the Université du Québec à Trois-Rivières (UQTR), QC, Canada. From 2017 to 2019, he was with Addvalue Innovation Private Ltd., Singapore, as a Satellite Communication Engineer. He worked on Linux-based software defined radio (SDR) platform development, where he was involved with board bring-up, FPGA prototyping and troubleshooting. His varied current research interests are baseband signal processing, approximate circuits, cyber-physical systems, computer vision, and machine learning.

Messaoud Ahmed Ouameur (Member, IEEE) received the bachelor's degree in electrical engineering from the Institute national d'électronique et d'électricité (INELEC), Boumerdes, Algeria, in 1998, the M.B.A. degree from the Graduate School of International Studies, Ajou University, Suwan, South Korea, in 2000, and the master's and Ph.D. degrees (Hons.) in electrical engineering from the Université du Québec à Trois-Rivières (UQTR), QC, Canada, in 2002 and 2006, respectively. From 2001 to 2006, he worked with Axiocom Inc., as the Director of research and development, where his research activities include wireless communications, spread-spectrum systems, iterative (turbo) detection, channel estimation, smart antennas, Monte Carlo techniques for signal processing, and real-time very-large-scale integration (VLSI). He joined Nutaq Innovation in November 2006. As a radio system technical leader, his tasks involved radio system design and performance analysis of wireless communication systems including GSM, WCDMA, LTE, and 5G, embedded signal processing algorithm design and implementation, and radio transceiver prototyping from the antenna to baseband (PHY) processing. He then joined UQTR as a Regular Professor in 2018. His research interests include the field of embedded real-time systems, parallel and distributed processing with applications to distributed Massive MIMO, deep learning and machine learning for communication system design, and the Internet of Things with the emphasis on end-to-end systems prototyping and edge computing.



Daniel Massicotte (Senior Member, IEEE) received the B.Sc.A. and M.Sc.A. degrees in electrical engineering and industrial electronics from the Université du Québec à Trois-Rivières (UQTR), QC, Canada, in 1987 and 1990, respectively, and the Ph.D. degree in electrical engineering from the École Polytechnique de Montréal, QC, Canada, in 1995. In 1994, he joined the Department of Electrical and Computer Engineering, Université du Québec à Trois-Rivières, where he is currently a Full Professor. He has been the Founder and the Head of the Laboratory of Signal and Systems Integration since 1998. Since 2001, he has been the Founding President and the Chief Technology Officer of Axiocom Inc. He was the Head of the Industrial Electronic Research Group from 2011 to 2018, the Head of the Department of Electrical and Computer Engineering from 2014 to 2020, and has been the Head of the Research Chair in Signals and Intelligence of High-Performance Systems since 2018. He has proposed many methods based on modern signal and biosignal processing, such as machine learning, transform domain, and metaheuristics. He has authored/coauthored more than 200 technical papers in international conferences and journals, and 9 inventions. His research interests include advanced VLSI implementation, digital signal processing for wireless communications, measurement, and medical and control problems for linear/nonlinear complex systems. He is a member of the "Ordre des Ingénieurs du Québec," "Groupe de Recherche en Électronique Industrielle (GREI)," and "Microsystems Strategic Alliance of Quebec (ReSMiQ)." He received the Douglas R. Colton Medal for research excellence awarded by Canadian Microelectronics Corporation, the PMC-Sierra High Speed Networking and Communication Award, and the Second place at the Complex Multimedia/Telecom IP Design Contest from Europractice. He was the General Chair of IEEE NEWCAS 2014 and a Guest Editor of *Analog Integrated Circuits and Signal Processing* (Springer) for the Special Issues of NEWCAS 2013.