International Journal of Engineering & Technology, 7 (4.33) (2018) 208-213

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Low Complexity Interference Alignment for Distributed Large-Scale MIMO Hardware Architecture and Implementation for 5G Communication

Savitri Galih*

Widyatama University, Bandung, Indonesia
**Corresponding author E-mail: savitri.galih@widyatama.ac.id*

Abstract

Massive MIMO or Large Scale MIMO is a promising solution for achieving superior data rates in 5G communication systems. However, it has limitation in term of scalability and coverage for users that has highly spatial separation. Distributed massive MIMO is expected to enhance these drawbacks. One main problem arises in this scheme is the MIMO interference channel condition that can be copied by interference alignment algorithm. The main consideration for interference alignment algorithm in distributed Massive MIMO is to achieve low complexity precoding to eliminate interference channel condition and to design efficient hardware architecture for its implementation. Previous research regarding IA for Distributed Massive MIMO indicate that the complexity issues is still not widely discussed. This paper proposed the low complexity IA scheme for large scale MIMO system based on limited interferer and the implementation of low cost interference alignment and wireless synchronization for distributed MIMO using software defined radio hardware. From the simulation result, it shows that limited interferer IA algorithm achieve acceptable BER performance, i.e. in order of 10^{-3}. The hardware implementation of the IA precoding matrix computation is also discussed. Based on the experiment, it is show that the proposed algorithm and architecture achieved higher hardware performance compared to the linear IA.

Keywords: *Distributed Massive MIMO; Interference Alignment; FPGA.*

1. Introduction

Delivering superior data rates without scarifying the quality of service is the key task for the wireless communication business. This task is well suited to the objectives of the 5G communication systems for achieving higher data rates. However, with the enormous growth of mobile data traffic due to the high number of mobile video streaming and mobile internet operation, it is challenging to manage the resulting condition, i.e. limited wireless spectrum and resources. One of the recommended solutions of this problem is Massive MIMO. Extensive research has been carried out since the first launch of Large Scale MIMO (Massive MIMO) to attain much greater scale of MIMO advantages i.e. (higher energy efficiency, higher data rate, higher reliability, and lower interference) [1]. Massive MIMO denotes the large number antenna fixed connected at base station. Despite of the advantages, massive MIMO has limitation in term of scalability and spatial diversity or coverage for highly spread user. To enhance these drawbacks, distributed Massive MIMO possibly will be a supreme way out. Distributed MIMO with physically distributed transmitters and receivers lead to the MIMO interference channel condition. Each firm one to one association between multiple transmitters and receivers will produce interferences between each transmission link. Recent research activities regarding analysis of the MIMO interference channel in topics of interference alignment has received significant attention among researchers. Interference Alignment is basically instituted by precoding methods for the MIMO interference channel [2]. The alignment process of the interference essentially is transmitted schemes that are constructed to make the interfering signals from transmitter is as low as possible [3-4].

The mentioned advantages of the large scale MIMO, Distributed MIMO and interference alignment, lead to the study in the implementation of interference alignment as a precoding system for Distributed Large Scale MIMO system. However, there are two main key consideration required for Distributed Massive MIMO with Interference Alignment (IA) realization, i.e. how to provide low complexity precoding to cope interference channel condition and to design accurate synchronization to enable coherent transmission.

Several recent studies in the topics of IA and/or precoding technique for large scale MIMO system [1, 5-7, 14] hardly emphasize the issue of computational complexity of the precoding/IA scheme. This paper will propose the low complexity IA scheme for large scale MIMO system based on limited interferer. We limit the interferer that contribute to the precoding matrix to reduce the complexity.

Moreover, the implementation of low cost interference alignment for distributed massive MIMO using Xilinx ISE Design platform is also discussed.

2. Review of Precoding Scheme for Large Scale MIMO System

Precoding process which employ an antenna array to lead transmitted signal directly to the intended receiver antenna is the core of Multi User MIMO. Precise Channel State Information is required for the precoding design in MU MIMO [8].

Since the beginning of large scale MIMO system, which has obtained great interest recently due to its numerous advantages in term of capacity, interference robustness, etc., there are some studies discussed about precoding process in this massive MIMO system. However, there are some issue regarding the computational complexity of IA and precoding schemes required for the system. It required excessive process to compute large scale precoding matrix in large scale MIMO system

Since distributed MIMO with its interference channel nature will be mainly discussed, in this work we focus primarily on low complexity interference alignment technique for large scale condition. There has been large amount of work on precoding and IA scheme for large scale antenna system. By exploiting polynomial expansion for rewriting the matrix inversion then truncate it, in [19] proposed low complexity linear precoding that can be implemented in large scale system. Despite, its function to eliminate precoding matrix precomputation and reduce transmitted symbol delay, the truncated polynomial expansion precoding implementation for Interference alignment need different approach that vastly difficult. For interference networks, in [10] developed two tier precoding with subspace alignment motivated by the clustering behavior of the user terminal. This scheme is implemented for classic (undistributed) massive MIMO with large number of antennas is placed in one centered place by considering one ring local scattering model. With such local condition requirement, this scheme's performance and feasibility is still unproved for implementation in distributed large scale MIMO system. In [16] proposed large scale system precoding algorithm by continually optimize receiving and transmitting precoding vectors iteratively based on Interference Alignment criterion and achieve significant performance gains. Despite its high performance result, this algorithm has not addressed high complexity issue which raise substantial obstacle in hardware implementation process.

3. Interference Alignment with Limited Interferer

3.1 System Model

Let us consider the large scale (16 rank and 32 rank) distributed MIMO interference channel with K user and N Number of antennas per user at each transmitter and receiver (Figure 1).

The received signal at the kth receiver at time t is represented as follows:

$$Y_k(t) = H_{k1}(t)X_1(t) + H_{k2}(t)X_2(t) + ... + H_{kK}(t)X_k(t) + Z_{k(t)} \qquad (1)$$

where, $k \in \{1,2,...,K\}$ is the user index, $Y_k(t)$ and $X_k(t)$ is the received signal at the kth receiver and transmitted signal from the kth transmitter respectively, $H_{kj}(t)$ is the frequency domain channel coefficient from transmitter j to receiver k at time t and $Z_k(t)$ is the additive white Gaussian noise (AWGN) term at receiver k. Some assumption is taken in the system i.e.: Channel State Information (CSI) at each node is perfectly known, all noise terms are independent identically distributed (i.i.d) zero mean complex Gaussian.

3.2 Linear Interference Alignment Review

In [4] discussed linear interference alignment algorithm in K-user MIMO interference channel with N receive and transmit antenna by employing symbol extension of the channel. In this condition, transmitted message is encoded by multiplication with IA precoding matrix V and the received signal at the ith receiver can be written as:

$$\mathbf{Y}_i(t) = \mathbf{H}_{i1}\mathbf{V}_1(t)\mathbf{X}_1(t) + \mathbf{H}_{i2}\mathbf{V}_2(t)\mathbf{X}_2(t) + \cdots + \mathbf{H}_{ij}\mathbf{V}_j(t)\mathbf{X}_j(t) + \mathbf{Z}_i(t) \qquad (2)$$

where $\mathbf{H}_{ij}$ is N x N frequency domain channel matrix between ith receiver and jth transmitter, $\mathbf{X}_j$ is symbol that is transmitted from jth node, and $\mathbf{Z}_i$ is zero mean AWGN vector at receiver i.

To suppress interference and reconstruct original transmitted signal, the received signal is decoded by suppression matrix U given by [11]:

$$\mathbf{U}_i = null\left(\left\lfloor \mathbf{H}_{ij}\mathbf{V} \right\rfloor_j\right) \qquad (3)$$

The linear IA algorithm carries out zero forcing to all undesired precoding vector at the receiver to eliminate interference. The interference vectors are linearly independent with the desired signal space.

3.3 Proposed IA

In this research, we aim to achieve low complexity Interference Alignment algorithm by proposing the limited interferer Interference Alignment for K-User MIMO.

3.3.1. Large Scale K-User SISO Interference Channel

For Large scale scenario, if the [4] Interference Alignment algorithm is used to determine V_1 will require high matrix dimension, i.e. if K = 32 and n = 1, the V1 matrix will consist of $2^N = 2^{(32-1)(32-2)-1}$. This condition lead to high complexity computation required for the IA process,

We modified IA precoding equation in 1st receiver V_1 from [4] in order to reduce the complexity in large scale system by limiting the maximum index of the equation. The cellular network phenomenon stated that the far-away interference signal power is highly decrease due to the distance between the non-adjacent users. Therefore, we can limit the range of the IA precoding equation to the number of significant interferer instead of the number of the user.

To determine the number of the significant interferer, we choose it based on modification of the theorem that characterize the feasibility of interference alignment stated below [12].

$$d \le \frac{2N}{L+1} \qquad (4)$$

where d = degree of freedom, N = Number of antenna per user and L = number of interferer. For N = 1 and d = 1, we get L = 3.

Based on the above assumption, for K user single input single output system, we limit the dimension of the interference alignment precoding equation's upper bond index from the number of user to the number of limited interferer (L) as shown in equation below:

$$\bar{V}_1 = \left\{ \left(\prod_{m,l} \left(\mathbf{T}_l^{[m]} \right)^{\alpha_{mk}} \right) \mathbf{w} : \forall \alpha_{m,k} \in \{0,1,2..,n\} \right\} \qquad (5)$$

where $m,l \in \{2,3,..,L\}, m \ne l, (m,k) \ne (2,3)$

$$\mathbf{T}_j^{[i]} = (\mathbf{H}_{i1})^{-1}\mathbf{H}_{ij}\mathbf{S}_j \qquad (6)$$

$$\mathbf{S}_j = (\mathbf{H}_{i1})^{-1}\mathbf{H}_{13}(\mathbf{H}_{23})^{-1}\mathbf{H}_{21} \qquad (7)$$

3.3.2. Large Scale K-User MIMO Interference Channels

In the large scale K-User MIMO interference channels systems, to decode the datastreams from the received signal vector, each undesired precoding vector space must have full intersection as follows:

At Receiver 1:

$$span(H_{12}V_2) = span(H_{13}V_3) = \cdots = span(H_{1K}V_K)$$

At Receiver 2:

$$span(H_{21}V_1) = span(H_{23}V_3) = \cdots = span(H_{2K}V_K)$$

At Receiver 3:

$$span(H_{31}V_1) = span(H_{32}V_2) = \cdots = span(H_{3K}V_K) \quad (8)$$

$$\vdots$$

At Receiver K:

$$span(H_{K1}V_1) = span(H_{K2}V_2) = \cdots = span(H_{KK-1}V_{K-1})$$

The above equations can be restricted with the Interference Alignment method described as

$$span(H_{12}V_2) = span(H_{13}V_3) = \cdots span(H_{1K}V_K) \quad (9)$$

$$\begin{aligned} H_{21}V_1 &= H_{23}V_3 = \cdots = H_{2K}V_K \\ H_{31}V_1 &= H_{32}V_2 = \cdots = H_{3K}V_K \quad (10) \\ &\vdots \\ H_{K1}V_1 &= H_{K2}V_2 = \cdots = H_{(K-1)K}V_K \end{aligned}$$

Based on [4], V$_i$, i = 2, 3, ..., K determined by

$$\begin{aligned} V_2 &= (H_{32})^{-1} H_{31}V_1 \\ V_3 &= (H_{23})^{-1} H_{21}V_1 \quad (11) \\ &\vdots \\ V_K &= \left(H_{(K-1)K}\right)^{-1} H_{K1}V_1 \end{aligned}$$

In (20a) and the assumption that the limited interferer is equal to 3 as in (4), we use same factor H$_{13}$V$_{13}$ to determine Vi, i = 4, 5, K below.

$$\begin{aligned} V_4 &= (H_{14})^{-1} H_{13}V_3 \\ V_5 &= (H_{15})^{-1} H_{13}V_3 \quad (12) \\ &\vdots \\ V_K &= (H_{1K})^{-1} H_{13}V_3 \end{aligned}$$

for determining V$_1$, we use the equation below.

$$span(V_1) = span(EV_1) \quad (13)$$

where

$$E = (H_{31})^{-1} H_{32} (H_{12})^{-1} H_{13} (H_{23})^{-1} H_{21}$$

We can calculate V$_1$ as the Eigen vector of E.

$$V_1 = eig[E]$$

3.3.2. Computational Complexity Discussion

With an increased number of antennas in large scale MIMO scheme, Inference Alignment process has to examine larger Interference Alignment precoding matrix (V).

In this section, the computational complexity of linear Interference Alignment algorithm from [4] and proposed IA is compared. The complexity is computed based on the number of multiplication required to compute Interference Alignment precoding matrix (V). By using same factor (h$_{13}$.v$_3$) for V$_i$ (i = 4 to K), we can reduce computational complexity since the factor only compute once and store it to memory and is read every Vi (i = 4 to K) computation, instead of compute it K – 3 times as in (12).

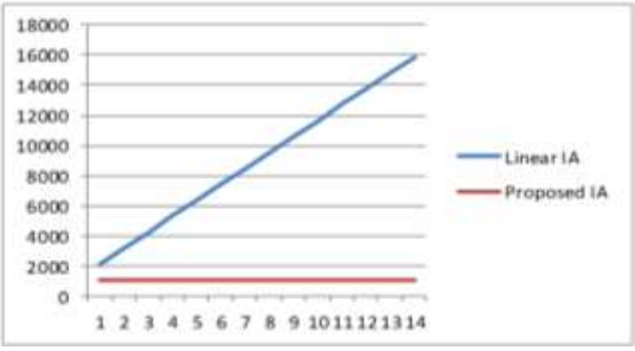

Fig. 1: Computational Complexity

For each i, i = 1 to K where K is the number of user with each M antenna, it requires one M rank matrix inversion and two M rank matrix multiplication. The computation will highly increase for large scale condition.

In Fig. 1 and Table 1 computational complexity comparison between Linear IA from [4] and proposed IA is shown.

Table 1: Computational Complexity

IA Algorithm	Multiplication #
Linear IA [4]	K.(M^2+M)
Proposed Algorithm	(M^2+M)

4. Simulation Results

In this section, we observe the bit error rate (BER) performance of proposed limited interference based IA algorithm in the large scale Distributed MIMO system and the effect of cost as loop synchronization in this large scale system. In our simulation, we set up two scenarios 16 and 32 distributed transmitters, each has 2 antennas and we also consider the case that the number of user (K) is 16 and 32. An i.i.d complex Gaussian distribution has built the channel matrix Hij with zero mean and unit variance. For all the simulation, we use QPSK modulation.

The simulation contains two part, i.e. one dedicated Masternode and 32 users distributed MIMO system. The Master node sends a reference signal to transmitters. By using this reference signal, the transmitters can estimate the offsets of each distributed transmitter. Costas loop then synchronize the signal by compensating the offsets. After synchronization process takes place, the transmitters carry out proposed limited interferer based IA algorithms, so that the transmitted signal can be retrieved without error due to interference in the desired receiver.

Table 2: BER Performance Difference after Sync

Number of User	BER Performance Difference (dB)
16	10 dB
32	13 dB

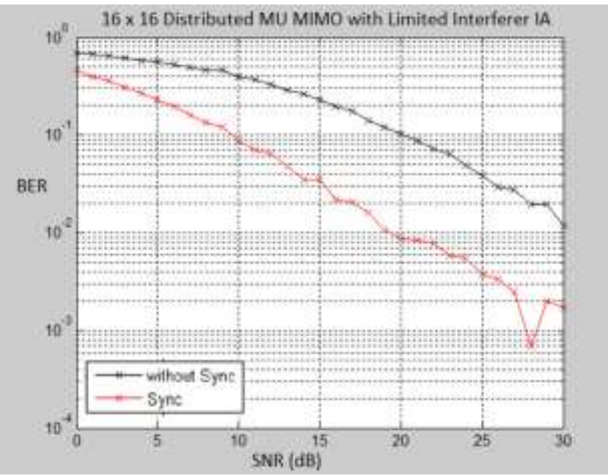

Fig. 2: BER performance of 16 x16 Distributed MU MIMO with Limited Interferer IA

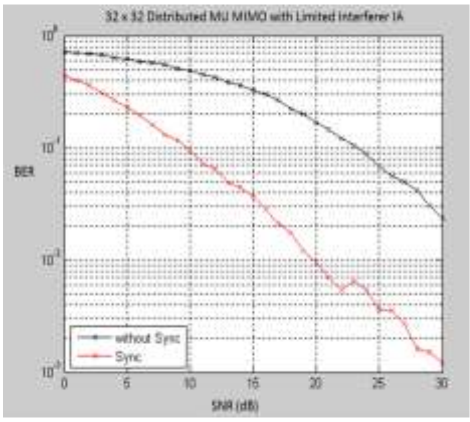

Fig. 3: BER performance of 32 x 32 Distributed MU MIMO with Limited Interferer IA

Figure 2 and 3 present the simulation of the synchronized and unsynchronized 16 and 32 users Distributed MIMO, which employ proposed limited interferer IA algorithm respectively. In this plot we can observe that the distributed synchronization algorithm increases the BER performance by 10 dB at axis (BER) value 10^{-1} for 16 users and 13 dB at BER value 10^{-1} for 32 users. We can conclude that the effect of the distributed synchronization in larger scale of distributed MIMO is higher than the less scale. The increasing of performance in larger scale distributed MIMO is due to higher number offset experienced by all transmitters that is compensated by the synchronization process.

Moreover, we can also observe the performance of the proposed IA algorithm. The simulation results show that the BER performance of the proposed limited interferer IA algorithm achieve acceptable BER performance (10^{-3}).

5. Hardware Implementation

In this section we discuss hardware implementation of limited interferer IA and the linear IA using VLSI technology.

4.1. Hardware Architecture

This section explains two architectures for calculating precoding matrix of IA in distributed large scale MIMO system. The first architecture implements the linear IA approach from [4] and the next architecture is mapped for proposed limited interferer method. The explanation of the two architecture enables us to compare the hardware performance of the two approaches.

The architecture for precoding matrix computation is shown in Figure 4. It consists of two main components, the controller unit and the Matrix Multiplication unit. The controller unit is basically finite state machine that performs matrix multiplication and memory storage.

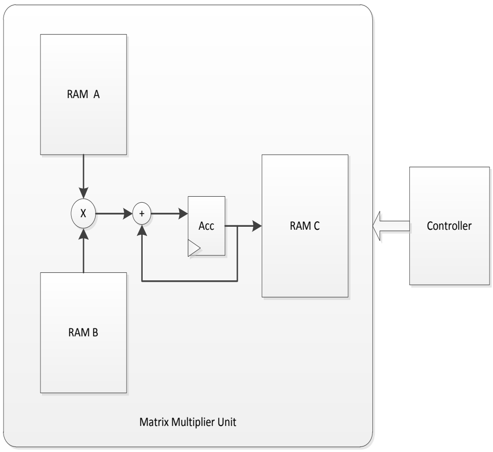

Fig. 4: Architecture of Limited Interferer IA Precoding Matrix Multiplication

The architecture of limited interferer (based on in (12)) is depicted in fig 4. The d the limited interferer IA only have one matrix multiplication module. On the other hand, the linear IA architecture have two matrix multiplication module. The proposed IA can reduce the number of matrix multiplication since it has preprocessing unit that compute another matrix multiplication only once, i.e. computes $H_{13}.V_3$ one time.

We used Xilinx ISE 14.7 and Xilinx Core Generator for designing the architecture, VHDL synthesis and place-and-route. The FPGA device for implementation target is Spartan 6 family XC6SLX4 device.

4.1.1. Architecture of Linear Interference Alignment

In Linear Interference Algorithm from [4], the interferer for every antenna index must be considered. Precoding Matrix (V_i, i = 2 to K) is determined in (11). The hardware architecture contains two matrix multiplier module and five memories. First Matrix Multiplier calculate $H_{i1}.V_1$ (i =2 to K). The second Matrix Multiplier calculate Vi (i = 2 to K). Both Matrix Multiplier must compute K-1 times to determine the precoding matrices.

4.1.2. Architecture of Limited Interferer Interference Alignment

The proposed hardware architecture (Figure 4) consists of three memories and one Matrix Multiplier module. The Matrix Multiplier module performs calculation to retrieve main IA precoding matrix in (12). For the Limited Interferer IA precoding matrix computation, we also need the inverse of channel coefficient matrix. The first input RAM stores the inverse of channel coefficient (H_{ij}^{-1}) and the second input RAM store the $H_{13}V_{13}$ calculation result. We assumed that channel coefficient matrix is already known and has been pre computed.

In order to achieve low complexity hardware implementation, we propose memory shared architecture that reduce the matrix multiplication computation. The number of the matrix multiplication process is reduced by precomputing factor in (12) one time and store it to one memory (RAM B). Hence, RAM B operates as shared memory accessed during matrix multiplication computation for every precoding vector space index. Since the $H_{13}V_{13}$ multiplication does not have to calculate everytime we compute each precoding matrix (V_4 to V_K), the computational complexity is highly reduced.

4.2 Finite State Machine (FSM)

To carry out the hardware implementation, the computation process is partitioned into a number of state. The FSM diagram of proposed algorithm is depicted in Figure 5.

4.2.1. Linear IA State

The state of the Linear IA algorithm is listed in the following. (SPT HUANG)

1) State stIdle: No activity, go to state stWriteBufferA01 if signal write enable and buffer select = 1
2) State stWriteBufferA01: Write data H_{k1} to RAM A01, go to state stWriteBufferA02 when counter reach 1024 (RAM size)
3) State stWriteBufferA02: Write data V_k to RAM A02 go to state stReadBufferA0 when counter reach 1024 (RAM size)
4) State stReadBufferA0: Read data H_{k1} and V_k for multiplying process, go to state stSaveDataA when counter A01 and counter A02 reach 32 (number of user/antenna).
5) State stSaveDataA: Ready to save the matrix multiplying result
6) stWriteBufferA: Write the matrix multiplying result to RAM A, go to state stWriteBufferB when counter reach 1024 (RAM size)
7) stWriteBufferB: Write data $H_{(K-1)K}$ to RAM B, go to state stWriteBufferB when counter reach 1024 (RAM size)

8) stReadBufferAB: Read data H_{k1} and V_k multiplying result and $H_{(K-1)K}$ for second multiplying process, go to state stSaveData when counter A and counter B reach 32 (number of user/antenna).
9) stSaveData: Ready to save the second matrix multiplying result
10) stWriteBufferC: Write the second matrix multiplying result to RAM C, go to state stComplete when counter reach 1024 (RAM size)
11) stComplete: k-index Precoding Matrix is ready, go to stIdle.

4.2.2. Limited Interferer State

The state of the Linear IA algorithm is listed in the following

1) State stIdle: No activity, go to state stWriteBufferA01 if signal write enable and buffer select = 1
2) stWriteBufferA: Write the precomputed H_{13} and V_3 to RAM A, go to state stWriteBufferB when counter reach 1024 (RAM size)
3) stWriteBufferB: Write data H_{1k} to RAM B, go to state stWriteBufferB when counter reach 1024 (RAM size)
4) stReadBufferAB: Read data H_{k1} and V_k multiplying result and $H_{(K-1)K}$ for second multiplying process, go to state stSaveData when counter A and counter B reach 32 (number of user/antenna).
5) stSaveData: Ready to save the second matrix multiplying result
6) stWriteBufferC: Write the second matrix multiplying result to RAM C, go to state stComplete when counter reach 1024 (RAM size)
7) stComplete: k-index Precoding Matrix is ready, go to stIdle.

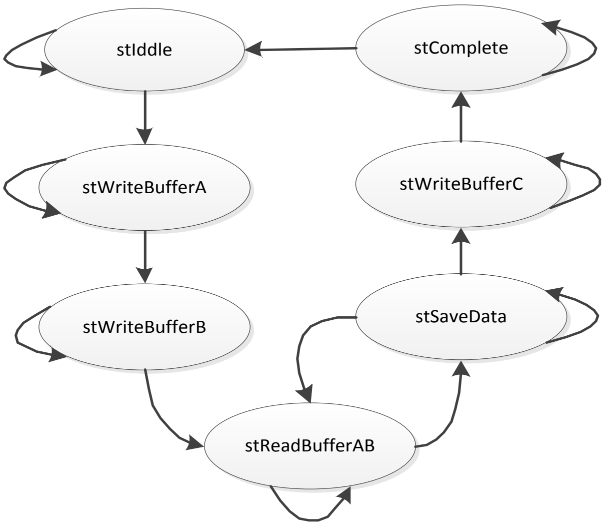

Fig. 5: Finite State Machine for Limited Interferer IA

4.3 Hardware Implementation Result

The design and development used Xilinx ISE 14.7 for VHDL synthesis and post place and route simulations. Both linear and limited interferer are implemented on Xilinx Spartan 6 FPGA. We compare the maximum frequency and hardware resource usage of the main IA precoding matrix in two algorithms. Based on the architecture described in section 4.1, the main precoding matrix calculation is mapped as customized hardware units on FPGA.

Table 3 summarizes the comparison of the post place and route frequency and the FPGA resource utilization for 32 x 32 MIMO System between linear IA and proposed Limited Interferer IA. The result show that the Linear IA Algorithm utilizes more multipliers, more FPGA Slices and more memory than Limited Interferer IA Algorithm, due to the Linear IA algorithm contains more matrix multiplication process.

The number slice of registers in Table 3 basically is used as flip flops representing finite state machine. Since Linear IA algorithm has more states than limited interferer IA, it uses more flip flops. Limited interferer IA reduced the number of slice register by 25 %. The number of slice LUTs can be acquired from the accumulation of the slice use as Logic (Arithmetic Logic Function), the number of slice used as memory and the number of slice used exclusively as route-thrus (routing paths between slices). The greatest part that contributes to the number of Slice LUTs in linear IA is the number used as memory, since it has more dual port RAM than proposed limited interferer IA. From the table, ot is shown that the number of slice LUTs is reduced by 21 %.

Table 3: FPGA Resource Usage for Linear IA and Limited interferer IA

	Linear IA Algorithm	Limited Interfere IA Algorithm
Target FPGA		
No. of Slices (Registers)	112 out of 93.296	84 out of 93.296
No. of Slices (LUT)	1299 out of 46.648	1023 out of 46.648
No. of Bonded IOB	100 out of 348	95 out of 348
No. of Multiplier	2	1
Maximum Frequency	85,008 MHz	80,942 MHz
Minimum Clock	11,7641 ns	12,355 ns

The second last row is the maximum frequency retrieved by the synthesis tool. It can be observed that Limited Interferer IA achieve lower maximum frequency by 4%.

Next hardware performance evaluation is carry out by estimating power consumption and area of the FPGA Implementation. The power consumption is estimated using the XPower Analyzer tool included in ISE. The power utilized by all component is accumulated using the capacitance model of the target device and the specific switching information (toggle rates, signal rates, and frequency information) to calculate the power consumption estimation.

The computation rate for IA algorithm, especially for precoding matrix calculation with N receive and transmit antennas is defined by [13].

$$C = \frac{N.f_{\max}}{clk_{avg}.n_{state}}$$

where $f_{\max}$ is frequency of the system in Megahertz, n_{state} is number of state and clk_{avg} is average number of clock cycle per state with bit per dimension is assumed equal to 1.

Table 4: Computation rate for Linear IA and Limited interferer IA

	Linear IA	Limited Interferer IA
Transmit/Receive Antennas	32	32
System Frequency	85 MHz	80 MHz
Number of State	11	7
Average Clock Cycle per State	4,162	4,623
Computation Rate	53,484 Mbps	87,869 Mbps

The simulation result exhibit that the average computation time per state is 4,623 clock cycle for Linear IA and 4,162 cycle for limited interferer IA. The computation rate of the two algorithm is given table 4. It is showed that the Limited Interferer IA algorithm is 1,64 times faster than Linear IA algorithm.

7. Conclusion

In this paper, we present the limited interferer based Interference Alignment algorithm and its precoding matrix system architecture and implementation for large scale distributed MIMO. In order to reduce the precoding matrix for large scale MIMO scenario, instead of use the user index to build the matrix, we limit it to the number of interferer that calculate by degree of freedom equation. Through the simulation result, it can be observed that the proposed limited interferer IA algorithm achieve acceptable BER performance.

We apply memory shared architecture to reduce the matrix multiplication computation since the shared memory store the precomputing factor one time so the matrix multiplication doesn't need to carry out every time we calculate the precoding matrix.

The system is prototyped on a Xilinx XC6SLX4-2-CSG75T device. It is showed that the Limited Interferer Algorithm has lower hardware complexity and has 1,64 times faster than Linear IA and use 21 % fewer FPGA LUT slices.

Acknowledgement

Part of this work has been published in a preliminary form in CEEC 2015.

References

[1] Larsson, E. G., Edfors, O., Tufvesson, F., & Marzetta, T. L. (2014). Massive MIMO for next generation wireless systems. IEEE Communications Magazine, 52(2), 186-195.

[2] Yang, S., & Hanzo, L. (2015). Fifty years of MIMO detection: The road to large-scale MIMOs. IEEE Communications Surveys and Tutorials, 17(4), 1941-1988.

[3] Schmidt, D. A., Utschick, W., & Honig, M. L. (2010). Large system performance of interference alignment in single-beam MIMO networks. Proceedings of the IEEE Global Telecommunications Conference, pp. 1-6.

[4] Cadambe, V. R., & Jafar, S. A. (2008). Interference alignment and degrees of freedom of the K-user interference channel. IEEE Transactions on Information Theory, 54(8), 3425-3441.

[5] Shepard, C., Yu, H., Anand, N., Li, E., Marzetta, T., Yang, R., & Zhong, L. (2012). Argos: Practical many-antenna base stations. Proceedings of the ACM 18th Annual International Conference on Mobile Computing and Networking, pp. 53-64.

[6] Suzuki, H., Kendall, R., Anderson, K., Grancea, A., Humphrey, D., Pathikulangara, J., Bengston, K., Matthews, J., & Russell, C. (2012). Highly spectrally efficient Ngara rural wireless broadband access demonstrator. Proceedings of the IEEE International Symposium on Communications and Information Technologies, pp. 914-919.

[7] Marzetta, T. L. (2010). Noncooperative cellular wireless with unlimited numbers of base station antennas. IEEE Transactions on Wireless Communications, 9(11), 3590-3600.

[8] Rogalin, R., Bursalioglu, O. Y., Papadopoulos, H., Caire, G., Molisch, A. F., Michaloliakos, A., Balan, V., & Psounis, K. (2014). Scalable synchronization and reciprocity calibration for distributed multiuser MIMO. IEEE Transactions on Wireless Communications, 13(4), 1815-1831.

[9] Mueller, A., Kammoun, A., Björnson, E., & Debbah, M. (2016). Linear precoding based on polynomial expansion: Reducing complexity in massive MIMO. EURASIP Journal on Wireless Communications and Networking, 2016(1), 1-22.

[10] Chen, J., & Lau, V. K. (2014). Two-tier precoding for FDD multicell massive MIMO time-varying interference networks. IEEE Journal on Selected Areas in Communications, 32(6), 1230-1238.

[11] El-Absi, M., El-Hadidy, M., & Kaiser, T. (2012). Antenna selection for interference alignment based on subspace canonical correlation. Proceedings of the IEEE International Symposium on Communications and Information Technologies, pp. 423-427.

[12] Bresler, G., Cartwright, D., & Tse, D. (2014). Feasibility of interference alignment for the MIMO interference channel. IEEE Transactions on Information Theory, 60(9), 5573-5586.

[13] Huang, X., Liang, C., & Ma, J. (2008). System architecture and implementation of MIMO sphere decoders on FPGA. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 16(2), 188-197.

[14] Cerato, B., & Viterbo, E. (2009). Hardware implementation of a low-complexity detector for large MIMO. Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 593-596.