ASIP Design for Multiuser MIMO Broadcast Precoding

Shahriar Shahabuddin, Olli Silvén, and Markku Juntti

***Abstract*—This paper presents an application-specific instruction-set processor (ASIP) for multiuser multiple-input multiple-output (MU-MIMO) broadcast precoding. The ASIP is designed for a base station (BS) with four antennas to perform user scheduling and precoding. Transport triggered architecture (TTA) is used as the processor template and high level language is used to program the ASIP. Several special function units (SFU) are designed to accelerate norm-based greedy user scheduling and minimum-mean square error (MMSE) precoding. We also program zero forcing dirty paper coding (ZF-DPC) to demonstrate the reusability of the ASIP. A single core provides a throughput of 52.17 Mbps for MMSE precoding and takes an area of 87.53 kgates at 200 MHz on 90 nm technology.**

***Index Terms*—MU-MIMO, Precoding, ASIC, ASIP, TTA.**

I. Introduction

Multiuser multiple-input multiple-output (MU-MIMO) is an advanced form of traditional single user MIMO where several users occupying the same bandwidth communicate with a base station (BS). MU-MIMO combines the benefits of traditional MIMO and space-division multiple access (SDMA) to provide significant improvements on the overall system performance [1].

Data detection in MU-MIMO is more complex traditional MIMO systems because different user receivers need to coordinate between themselves to cancel the interference. If the channel information is available at the transmitter, it is possible to remove the interference by precoding techniques. Another advantage of precoding techniques is the possibility to simplify the receivers [2]. On the other hand, user scheduling is also necessary for a practical MU-MIMO system due to the limited number of antennas at the BS. The performance of user scheduling and precoding depends on each other and both can affect the overall system performance. It is very difficult to select a particular user scheduling or precoding algorithm as there are numerous algorithms available for different scenarios. There is a growing need for flexible implementation of the MU-MIMO precoders with the capability to update whenever necessary.

Several hardware implementations for MU-MIMO precoding are proposed in [3], [4] and [5]. The fixed hardware implementations provide high data rate and consume less silicon area compared to the customized application specific processors (ASIP). The drawback of the fixed hardware implement ation is that it operates only on a fixed set of parameters due to the hardwired control path and it is very difficult to modify the control path in the future. An ASIP customized for a small set of algorithms is an attractive solution in terms of cost, silicon area and high throughput. Most importantly, an ASIP reduces the design risk with an instruction memory that can be used to load new programs or control instructions.

In this paper, we propose an ASIP for MU-MIMO broadcast precoding. The ASIP is based on the transport triggered architecture (TTA) paradigm. TTA is a processor design philosophy where the programmer can control the internal data transports between different function units of the processor. TTA exploits the instruction level parallelism (ILP) by processing several instructions in a single clock cycle [6], [7].

A norm-based greedy scheduler is used in this work. The scheduler selects four user indices out of a total twenty users. QR decomposition on augmented channel matrix is used to simplify and solve the MMSE filtering problem. We also configured the ASIP with high level language to support low complexity zero forcing dirty paper coding (ZF-DPC). The precoder design is more realistic as it considers scheduling unlike most other precoder implementation. To our knowledge, this is the first precoder ASIP based on TTA architecture. The ASIP can perform greedy scheduling, QR decomposition and MMSE filtering in 102, 340 and 92 clock cycles respectively. The architecture is synthesized on 90 nm technology and takes an area of 87.53 kgates at 200 MHz.

The rest of the paper is organized in the following way: The system model and precoding algorithms are presented in Section II and Section III. The processor architecture is presented in Section IV. In Section V, the simulation results and the processor performance are discussed. The conclusion is drawn in Section VI.

II. System Model

We consider a single cell downlink channel with a M antenna BS serving a total N single antenna users. The set $\mathcal{U}$ consists of the integer indices of all users in the system. At any given instant, the BS transmits data for a subset $\mathcal{A} \subset \mathcal{U}$ where $|\mathcal{A}| = M$. $\mathcal{A}$ is the active user set that consists of the indices of the multiplexed users at a given schduling instant. $\mathcal{A}$ is selected by greedy scheduling where the norm of all user channels are calculated and M users with highest norm are selected such that

$$\mathcal{T} = \arg \max_{\forall \mathcal{T} \in \mathcal{U} \setminus \mathcal{A}} \|\mathbf{h}_k\|^2. \tag{1}$$

S. Shahabuddin and M. Juntti was with Centre for Wireless Communications, University of Oulu, Oulu - 90570, Finland, e-mail: firstname.lastname@oulu.fi

O. Silvén is with Department of Computer Science and Engineering, University of Oulu, Oulu - 90570, Finland, e-mail: firstname.lastname@oulu.fi

This research work was supported by Academy of Finland.

We initialize the active user set as an empty set, $\mathcal{A} = \{\phi\}$.

The BS transmits to M different active users through M antennas at any time instant. However, the transmitted signals for different users interfere with each other and thus corrupt the signal designated to any particular user. Thus, the received signal for user k can be expressed as

$$y_k = \mathbf{h}_k^H \mathbf{x}_k + \sum_{j \neq k} \mathbf{h}_k^H \mathbf{x}_j + n_k, \tag{2}$$

where $\mathbf{h}_k \in \mathbb{C}^{M \times 1}$ is the channel vector between the BS and user k, $\mathbf{x}_k \in \mathbb{C}^{M \times 1}$ is the transmitted signal for user k and n_k is zero mean Gaussian noise.

The transmitted vector for user k is obtained by multiplying the beamforming vector $\mathbf{w}_k$ and symbol u_k as

$$\mathbf{x}_k = \mathbf{w}_k u_k. \tag{3}$$

The beamforming vector $\mathbf{w}_k$ is applied to avoid the interference caused by other transmitted signals.

We stack the channel vectors to form a channel matrix $\mathbf{H} \in \mathbb{C}^{M \times M}$ and beamforming vectors to form the precoding matrix $\mathbf{W} \in \mathbb{C}^{M \times M}$ and thus the input-output relation can be written as

$$\mathbf{y} = \mathbf{H}\mathbf{W}\mathbf{u} + \mathbf{n}, \tag{4}$$

where $\mathbf{u}$ is a vector of the original symbols , $\mathbf{n}$ is the noise vector and $\mathbf{y}$ is the received signal vector.

Typically, precoders are designed with respect to a total power constraint of the form

$$E\|\mathbf{x}\|^2 = \text{Tr}\{\mathbf{W}\mathbf{W}^H\} \leq P, \tag{5}$$

where total power, $P > 0$. Total power constraint simplifies the design problem and leads to simple precoders.

III. Precoding Algorithm

A. Zero-Forcing Precoding

Zero forcing (ZF) is one of the simplest precoding methods. The multiuser channel is decoupled to multiple independent sub-channels in the ZF precoding. ZF can perform very well when the signal-to-noise ratio (SNR) or the number of users is sufficiently high. ZF precoder is essentially a channel inversion problem. Wiesel *et al.* have shown that pseudo-inverse based precoder is optimal for maximizing any performance measure under total transmit power constraint [2]. ZF precoding matrix can be expressed as

$$\mathbf{W}_Z = \mathbf{H}^H(\mathbf{H}\mathbf{H}^H)^{-1}. \tag{6}$$

B. MMSE Precoding

ZF precoders do not provide linear capacity growth in the multi user channel. This is due to the fact that ZF precoders impose a stringent requirement that the interference from the receivers has to be removed [8]. A small amount of interference at each receiver helps to consider a large set of potential solutions that provide higher capacity for a given transmit power [1]. We use a regularization of the pseudo-inverse to compute the MMSE precoding matrix as

$$\mathbf{W}_M = \mathbf{H}^H(\mathbf{H}\mathbf{H}^H + \alpha^2\mathbf{I})^{-1}. \tag{7}$$

where α^2 is the regularization factor. A non-zero regularization factor can be used to allow a measured amount of multi-user interference.

C. MMSE Precoding with QR Decomposition

An extended channel matrix can be formed to solve the MMSE problem in the following way

$$\underline{\mathbf{H}} = \begin{bmatrix}\mathbf{H} & \alpha\mathbf{I}_N\end{bmatrix} \Leftrightarrow \underline{\mathbf{H}}^H = \begin{bmatrix}\mathbf{H}^H \\ \alpha\mathbf{I}_N\end{bmatrix}. \tag{8}$$

The right pseudo-inverse of the extended channel matrix takes the following form where the upper half of the equation is the same as MMSE precoder expression of (6).

$$\underline{\mathbf{W}} = \begin{bmatrix}\mathbf{W}_M \\ \alpha(\mathbf{H}\mathbf{H}^H + \alpha^2\mathbf{I})^{-1}\end{bmatrix}. \tag{9}$$

We apply QR decomposition on the Hermitian transpose of extended channel matrix $\underline{\mathbf{H}}$ as

$$\underline{\mathbf{H}}^H = \begin{bmatrix}\mathbf{H}^H \\ \alpha\mathbf{I}_N\end{bmatrix} = \underline{\mathbf{Q}\mathbf{R}} == \begin{bmatrix}\mathbf{Q}_1\underline{\mathbf{R}} \\ \mathbf{Q}_2\underline{\mathbf{R}}\end{bmatrix}. \tag{10}$$

The inversion of $\mathbf{R}$ can be easily found by

$$\alpha\mathbf{I}_N = \mathbf{Q}_2\underline{\mathbf{R}} \Leftrightarrow \underline{\mathbf{R}}^{-1} = \frac{1}{\alpha}\mathbf{Q}_2. \tag{11}$$

Afterwards we use the expression of (10) to further simplify (11) as

$$\underline{\mathbf{W}} = \frac{1}{\alpha}\underline{\mathbf{Q}}\mathbf{Q}_2^H = \frac{1}{\alpha}\begin{bmatrix}\mathbf{Q}_1\mathbf{Q}_2^H \\ \mathbf{Q}_2\mathbf{Q}_2^H\end{bmatrix}. \tag{12}$$

From (9) and (12) we get

$$\mathbf{W}_M = \frac{1}{\alpha}\mathbf{Q}_1\mathbf{Q}_2^H. \tag{13}$$

A similar approach is taken in [9] to simplify the MMSE precoding applying QR on extended channel matrix. The regularization factor is traditionally calculated as

$$\alpha^2 = \frac{M\sigma^2}{P}, \tag{14}$$

where σ^2 is the noise variance and P is the power constraint.

D. ZF-DPC Precoding

Dirty paper coding (DPC) is a highly nonlinear technique and its implementation is a very challenging problem [10]. Zero forcing dirty paper coding (ZF-DPC) is a reduced complexity suboptimal DPC scheme that was first proposed in [11]. The channel matrix is decomposed to a lower triangular matrix $\mathbf{L} \in \mathbb{C}^{M \times M}$ and a unitary matrix $\mathbf{Q} \in \mathbb{C}^{M \times M}$ to apply the ZF-DPC. It converts the symbol vector such a way that multiplying the symbol vector with $\mathbf{L}$ creates a diagonal matrix [12]. Afterwards, the modified symbol vector is multiplied by Hermitian transpose of the unitary matrix, $\mathbf{Q}^H$ and transmitted over the channel. A new symbol vector $\tilde{\mathbf{u}}$ to convert the non-diagonals of $\mathbf{L}$ to zero can be obtained as

$$\tilde{u}_i = u_i - \sum_{j=1}^{j=i-1} \frac{l_{ji}}{l_{ii}} u_j, \tag{15}$$

where $\mathbf{u}$ is the original symbol vector. ZF-DPC pre-cancels the interference without any loss of information.

IV. MU-MIMO Precoding ASIP

The ASIP is designed for a BS with $M = 4$ antennas that serves M active users out of a total $N = 20$ users. The precoder chain can be divided in four sections as shown in Fig. 1, they are scheduling, matrix decomposition, precoding and power constraint. A norm-based greedy scheduling and total power constraint is used in this work. MMSE precoding is the primary focus of this work, but the ASIP is designed in such a way so that it can support ZF-DPC too. QR-decomposition is used for matrix decomposition as it is needed for both precoding algorithms.

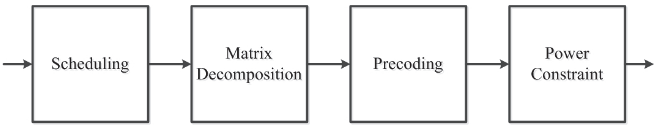

Fig. 1. MU-MIMO precoder.

A. Special Function Units

Several hardware units to accelerate MU-MIMO precoding are designed for the processor in addition to the general purpose function units (FU). We call them special function unit (SFU) throughout the paper. Two SFUs are designed to accelerate the greedy scheduling. A SFU called MGN is designed to calculate the absolute value of any complex number. Two real valued multipliers are used inside the MGN to compute the square of real and imaginary parts of the inputs. Another SFU, named SORT, is designed for sorting the values. We use an insertion sorter that takes the summation of absolute values and the corresponding indices as inputs at a time and keeps the indices of highest four values in sorted order.

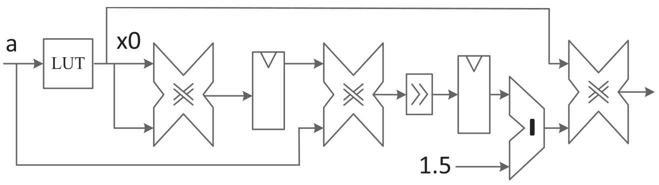

Fig. 2. Inverse square root (ISQRT) SFU.

The reciprocal of the norm of a vector is needed in QR decomposition. We design a three cycle inverse square root unit called ISQRT in this work. The architecture of the ISQRT unit is shown in Fig. 2. A look-up table (LUT) is used to hold the precomputed inverse square root values of all possible integers of the fixed point input. A 6-bit integer is used for the fixed-point input and thus, a LUT of size 2^6 is used for ISQRT. The output of the LUT x_0 is used as an initial guess for the Newton-Rhapson method. A single iteration of the Newton-Rhapson is used to find the square root of any input a as

$$x_1 = x_0(1.5 - .5 * a * (x_0)^2). \tag{16}$$

Three real multipliers are used in ISQRT and two registers are used in between to shorten the critical path. A similar approach is taken to design a real-valued division circuit that is needed for ZF-DPC precoding.

B. High Level Architecture

A part of the TTA processor for MU-MIMO precoding is illustrated in Fig. 3. For readability, the whole processor figure is not given. The black horizontal straight lines represent the buses of the processor. The vertical rectangular blocks represent the sockets.

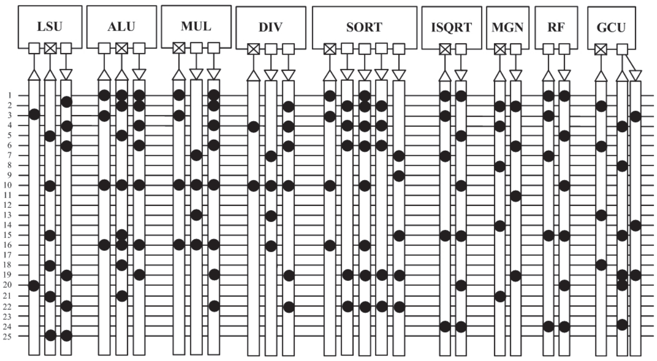

Fig. 3. Implemented processor with reduced number of functional units.

The 32-bit fixed point processor core includes the load/store unit (LSU), arithmetic logic unit (ALU), global control unit (GCU), register files (RF), several SFUs and conventional FUs. The channel vectors of $N = 20$ users are stored in a memory. LSU can read a single value from the memory in a single cycle. A first-in-first-out (FIFO) memory buffer is used to write the output of the processor. Due to the data dependency of QR algorithm, it is not possible to utilize more than four multipliers in an instant. Therefore, four complex-multipliers are included in the ASIP. The real division unit is used for the ZF-DPC calculations of (15). Twenty five buses are used to support ILP and reduce the latency. Fifteen RFs are used to save the intermediate results in this work. The GCU is used to support jump and branching.

V. Results and Discussion

We present sum-rate performance of greedy scheduling and compare with Gram-Schmidt (GS) based semiorthogonal user selection (SUS) and Round Robin scheduling algorithm in Fig. 4. ZF precoding is used with all schdulers. A cell edge scenario is considered where path loss is assumed to be equal for all the users in the system in order to eliminate the bias. By doing so, we ensure that the sum rate performance in Fig. 4 is only guided by the schedulers. The greedy scheduler achieves lower sum rate compared to SUS, but achieves higher sum rate compared to Round Robin scheduler. However, its complexity is significantly lower than SUS. Moreover, the gap between greedy and SUS decreases when path loss is introduced.

We present the bit error-rate (BER) performance of ZF, MMSE and ZF-DPC precoders for various SNR in Fig. 5. An additive white Gaussian noise (AWGN) channel is used for QPSK modulation and the BER is averaged over 100 000 Monte-Carlo trials. A greedy scheduler is used to select four users out of $N = 20$ users. It can be seen that the BER performance of DPC is better than MMSE and ZF in this scenario.

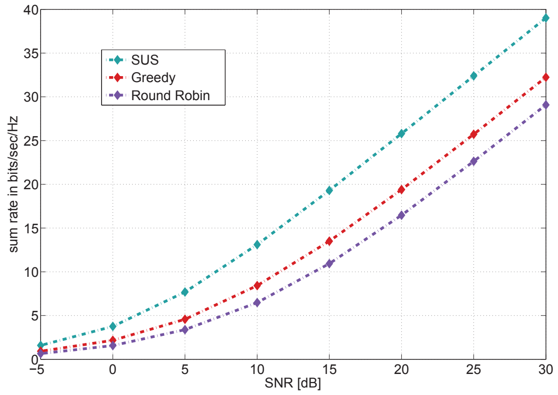

Fig. 4. Performance of different schedulers.

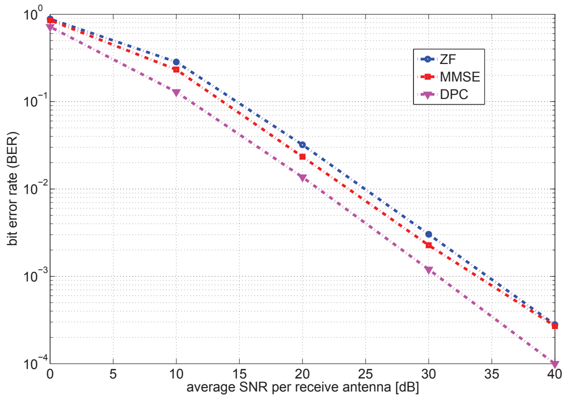

Fig. 5. Performance of different precoders.

The TTA is configured with C code and macros are used to call the SFUs. The clock cycle needed to execute scheduling, matrix decomposition and precoding is provided in the Table I. Memory access is a costly operation and can increase the latency significantly. For example, the scheduler needs to access the memory 80 times to read channel vectors of $N = 20$ users. QR decomposition also needs to access memory frequently and thus increase latency. The number of clock cycles needed for MMSE and DPC are nearly equal. However, DPC needs an extra division unit.

A very small number of hardware implementation can be found for MU-MIMO precoding. A comparison with different other precoder implemntations of MU-MIMO precoder is presented in Table II. A DPC precoder based on nested trellis is implemented on FPGA in [4]. A Tomlinson-Harashima (TH) precoder is designed in [5] where the LQ decomposition is implemented in ASIP and the other parts are implemented as monolithic hardware.

The proposed ASIP performs very well in terms of throughput compared to [4] and [5]. In addition, the proposed ASIP is more realistic as it considers scheduling unlike the other implementations. The flexibility of the ASIP is also demonstrated with another different precoding algorithm, ZF-DPC. The costly memory access can be removed with a multiprocessor or vector architecture, but it will also increase the area significantly.

TABLE I
LATENCY OF DIFFERENT PARTS OF THE PRECODER CHAIN

Algorithm	Clock Cycle
Greedy Schduling	102
QR	340
MMSE	92
DPC	97

TABLE II
IMPLEMENTATION COMPARISON

Reference	Architecture	Algorithm	Throughput
Propsed	TTA ASIP	MMSE	52.17 Mbps
[5]	ASIP & VLSI	TH	N/A
[4]	Virtex	DPC	51 Mbps

VI. CONCLUSION

We propose an ASIP for MU-MIMO scheduling and precoding. We simulate the performance of the scheduler and precoder in Matlab environment and propose a customized TTA processor. The processor is programmable with a retargetable compiler. The ASIP can support multiple precoding algorithm and can achieve low latency.

REFERENCES

[1] Q. H. Spencer, C. B. Peel, A. L. Swindlehurst, and M. Haardt, ”An introduction to the multi-user MIMO downlink”, in *IEEE Comm. Magazine*, vol. 42, no. 10, pp. 60-67, Oct 2004.

[2] A. Wiesel, Y. C. Eldar, and S. Shamai, ”Zero-forcing precoding and generalized inverses”, in *IEEE Trans. Signal Process.*, vol. 12, no. 3, Mar 2013.

[3] A. Burg, D. Seethaler, and G. Matz, ”VLSI Implementation of a Lattice-Reduction Algorithm for Multi-Antenna Broadcast Precoding”, *IEEE Intl. Conf. of Ckts. and Sys. (ISCAS)*, pp 673-676, New Orleans, LA, May 2007.

[4] P. Bhagawat, W. Wang, M. Uppal, G. Choi, Z. Xiong, M. Yeary, and A. Harris, ”An FPGA Implementation of Dirty Paper Precoder”, *Intl. Conf. of Comm. (ICC)*, pp 2761-2766, Jun 2007.

[5] K. Shimazaki, S. Yoshizawa, Y. Hatakawa, T. Matsumoto, S. Konishi, and Y. Miyanaga, ”A VLSI design of an arrayed pipelined Tomlinson-Harashima precoder for MU-MIMO systems”, *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. SummitConf. (APSIPA)*, pp 1-4, 2013.

[6] P. Jääskeläinen, V. Guzma, A. Cilio, T. Pitkänen, and J. Takala, “Codesign toolset for application-specific instruction-set processors,” in *Multimedia on Mobile Devices 2007*, vol. 6507 of *Proceedings of SPIE* pp. 1-11, San Jose, Calif, USA, Jan 2007.

[7] P. Salmela, H. Sorokin, and J. Takala, ”A programmable max-log-MAP turbo decoder implementation,” *Hindawi VLSI Design*, vol. 2008, pp 636-640, 2008.

[8] C. B. Peel, B. M. Hochwald, and A. L. Swindlehurst, ”A vector- perturbation technique for near-capacity multiantenna multiuser communication- part I: channel inversion and regularization”, *IEEE Trans. on Comm.*, vol. 53, no. 1, pp. 195-202, Jan 2005.

[9] C. W. Chen, H. W. Tsao, and P. Y. Tsai, ”Equal-rate QR decomposition based on MMSE technique for multi-user MIMO precoding” in *Proc. IEEE Int. Symp. Pers., Indoor, Mobile Radio Commun.*, pp. 435-440, 2013.

[10] A. D. Dabbagh and D. J. Love, ”Precoding for multiple antenna Gaussian broadcast channels with successive zero-forcing,” in *IEEE Trans. on Signal Proc.*, vol. 55, no. 7, Jul 2007.

[11] G. Caire and S. Shamai, ”On the achievable throughput of a multiantenna Gaussian broadcast channel,” in *IEEE Trans. on Inf. Theory*, vol. 49, no. 7, Jul 2003.

[12] L. Tran, M. Juntti, and M. Bengtsson, ”Beamformer design for MISO broadcast channels with zero-forcing dirty paper coding,” in *IEEE Trans. on Wireless Comm.*, vol. 12, no. 3, Mar 2013.