ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Low complexity Pipelined Implementation of Vector Precoding for MIMO systems

M. Bala Krishnan[*1], S. Sobana[2], Dr. K. Meena[3]
[*1] PG Student, [2] Associate Professor, [3] Professor, Department of ECE, PSNA College of Engg and Tech, Tamilnadu, India
balaece.com@gmail.com

Abstract

The nonlinear vector precoding (VP) technique has been used to achieve the capacity performance in multiuser multiple input multiple-output (MIMO) downlink channels. The performance promote with respect to its linear counterparts stems from the combination of a disturbed signal that reduces the power of the precoded signal. However, the optimality of these algorithms has been compared mainly in terms of error-rate performance and computational complexity, leaving the hardware cost based on their implementation an open distribute. The proposed work improves its performance in terms of slices, flip flops, latency, comparators, gate counts and power consumption. This paper has addressed the issues of a fully-pipelined implementation of the FSE and K-best architecture approaches for a 4 × 4 VP system.

Keywords - Vector Precoding (VP) , non-linear VP, FPGA, MIMO, perturbation.

Introduction

The representation of the vector precoding (VP) technique [1]is used for data communication over the multiuser broadcast channel, several algorithms have been proposed in the literature survey carrying the information to replace the computationally stubborn complete search defined in the original description of the algorithm. To this respect, frame work reduction approaches have been generally used as a means to measure a suboptimum perturbation vector with a arbitrate complexity. The key idea of framework reduction techniques depend on the usage of an equivalent and more advantages set of basis vectors to allow for the suboptimal resolution of the full or complete search problem by means of a simple rounding operation. This method is used in [2], where the Lenstra-Lenstra-Lov´asz (LLL) reduction algorithm[3] is used to measure the Babai's approximate closest point solution [4]. Similar methods can be found in [5]. Despite achieving complete diversity order in VP systems [6], the performance degradation caused by the quantization error due to the rounding operations. Moreover, several framework reduction algorithms have a considerable computational complexity, which poses several challenges to a future hardware implementation. An appropriate perturbation vector can also be found by searching for the optimum solution within a subset of candidate vectors. This approaches, is also called as tree-search techniques, perform a traversal through a trees which demand with the aim of finding a suitable perturbation vector. Even though it has the high volume of research work publish around the topic of precoding algorithms, the problem raised by their implementation have not been given the same attention. Some of the insufficient publications about this area, such as describe in [7], precoding systems that has either a considerable complexity in terms of allocated hardware resources or provide a rather low bit rate transmission.Even there is less number of published research in the area of hardware architectures for precoding algorithms, the implementation distribute of tree-search schemes in MIMO detection scenarios have been extensively studied. For example, the field programmable gate array (FPGA) implementation of the fixed-complexity sphere decoder (FSD) detector has been analyzed in [8-11], whereas the hardware architecture of the *K*-best tree search algorithm considering a real equivalent model was researched in [12]. Moreover, the implementation of suboptimum complex-plane enumeration and *K*-best detector was performed in [13-14]. A complete review of *K*-best tree-search technique implementation was carried out in[15].The adaptation of these tree-search schemes to precoding systems implies several variations with respect to the algorithms of the original description. Even if many lessons can be learned from the hardware architecture of tree-search techniques for point-to-point MIMO systems, the peculiarities of the precoding scenario render the results of the aforementioned publications inadequate for the current research topic. Consequently, this contribution addresses the high throughput implementation of fixed-complexity tree-search algorithms for VP systems. More specifically, two state-of the- art tree-search algorithms that allow for the parallel processing of the tree branches have been implemented on a Xilinx Virtex VI FPGA following a rapid-prototyping methodology. In order to achieve a high throughput, both schemes operate in the complex plane and have been implemented in a fully-pipelined fashion providing one output per cycle.

General Architecture Overview

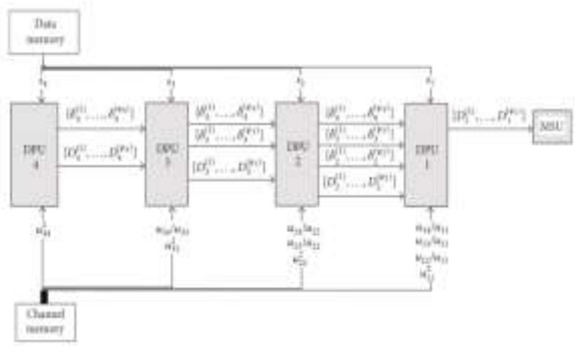

Fig 1 General hardware architecture

The same general distance computation structure, as can be seen in Figure 1. The lack of loops in the hardware architecture of the fixed-complexity tree-search techniques enables a high throughput and fully pipelined implementation of the data process, thus being its implementation specially suitable for a target FPGA device. The AEDs of the candidate branches are computed by accumulating the PEDs calculated at the local distance processing units (DPUs) to the AEDs of the previous level. This way, the AEDs down to level i corresponding to the considered candidate branches, namely, [$D^{(1)}{}_i$, . . . , $D^{(\psi i)}{}_i$], are passed on from DPU i to DPU $i-1$.The parameter ψ_i stands for the number of candidate branches at each level of the tree search, being it $\psi_i =$ K for all i for the K-best and $\psi^i = \prod_{j=1}^{N} n_j$ for the FSE model. Two input memory blocks, named Data Memory and Channel Memory, have been included to store the data symbols and the values of the triangular matrix **U**, respectively. The computation of the intermediate points z_i requires the values of all previous $\delta_j = a_j + s_j$. To avoid redundant calculations, the set of values [$^{(1)}$, . . . , $\delta^{(\psi j)}{}_j$] for all $j > i$ is transferred to DPU i, as is shown in Fig 1.

Proposed Architecture Design

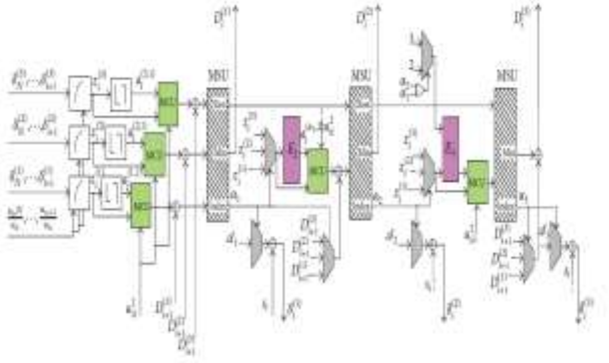

Fig 2 Architecture of K-best DPU

The structure of the proposed K-best DPU is depicted in Figure 2 for a system with K = 3. The branch selection procedure is carried out in K fully-pipelined sorting stages following a modified version of the WPE algorithm . First of all, the computation of the intermediate points is performed for each one of the K branches that are passed on from the preceding level. The set of best child nodes that stem from each parent node can be computed by simply rounding off the value of the intermediate point to the nearest lattice point. The distance increments (d_i in (2)) for those$K$-best children are computed by K metric computation unit (MCU) and are accumulated with their corresponding D_{i-1} values. These distance values and their corresponding branches comprise the candidate list A_1.Theminimum AED within A_1 is found at the minimum search nit (MSU) by simple concatenation of compare-and choose blocks. The MSU also outputs the index of the first winner branch $\alpha_1 \in \{1, . . . , K\}$ so that the appropriate value of z_i can be selected for the local enumeration procedure. zi can be selected for the local enumeration procedure. At the second stage of the sorting procedure, the $a_i^{((\alpha 1,2)}$ node needs to be identified for any parent node index α_1.This task is performed by the $E2$ block, which comprises a puzzle enumerator that outputs the second most favorable node given a certain value of z_i. However, in the subsequent stages of the algorithm, the enumeration procedure will depend on the index of the previously appointed winner branches. Hence, if $\alpha_1 = \alpha_2$, the third most promising child node will need to be expanded, namely, $a_i{}^{(\alpha 1,3)}$, whereas the second most favorable node in the $\alpha 2$ branch ($a_i^{(\alpha 2,2)}$) will be required. if $\alpha_1 \neq \alpha_2$ Consequently, the new candidate branch to be included in the A$_k$ candidate list at the k^{th} sorting stage will require the expansion of the ρ^{th} most favorable child node, where ρmay take any value within the set $\{2, 3, . . . , k\}$. The enumeration approach at each sorting stage k has been carried out by means of a puzzle enumerator unit capable of ascertaining the optimum ordered sequence of the first k child nodes in a non sequential fashion. The node order determination in the puzzle enumerator can be carried out without performing any costly distance computations. For each sorting stage k any node in the ordered sequence of best k child nodes can be selected for expansion. This way, the preferred child node in the ordered sequence is determined by an additional input variable which keeps track of the amount of already expanded child nodes for each parent node. The puzzle enumerator has been chosen as the enumeration scheme to be used along with the WPE due to its lower hardware resource demand and non sequential nature. Note that, there are no feedback loops in the structure of the *K*-best DPU, and therefore, it is possible to implement it following a fully-pipelined scheme.

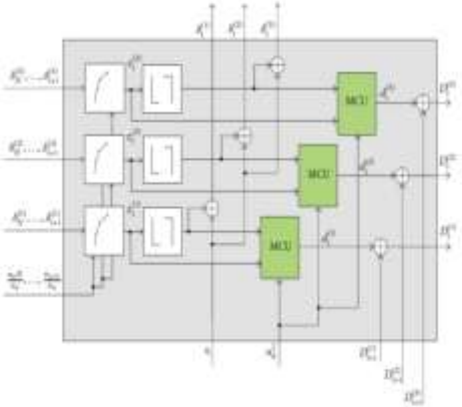

Fig 3. FSE DPU structure

The intricate node ordering and selection procedure required by the *K*-best algorithm is replaced by a simple Schnorr-Euchner enumerator in the FSE tree-search model. This derives in a considerably simpler DPU architecture of the FSE scheme. Figure 3 depicts the structure of the FSE DPU, where the block diagram for $n_T^i = \prod_{j=1}^{i-1} n_j$ =3 and n_i = 1 is represented. First of all, the data of the $\{\delta_{i+1}, . . . , \delta_n\}$ values transferred from level i + 1 are used to compute the intermediate values z_i for each one of the parent nodes. Afterwards, the node selection procedure is performed by means of a simple rounding operation when n_i = 1, as depicted in the illustrative example in Figure 3, or by means of the unarranged puzzle enumerator for the cases where n_i > 1.The PEDs of the selected nodes are then computed by $n_T^{(i)}$ MCUs and accumulated to the AEDs from the preceding level. Finally, as was the case with the *K*-best DPU, the FSE DPU does not have any feedback paths in its design, and hence, it can be easily implemented following a fully pipelined scheme.

Simulation Results

The proposed sphere decoder architecture is designed using verilog HDL, simulated using model sim software and synthesized using Xilinx project navigator. The RTL schematic view is illustrated in fig 4 and its technology schematic view is displayed in fig 5.

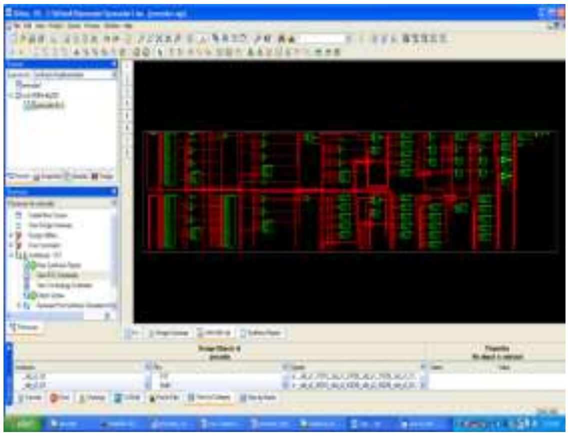

Fig 4. RTL Schematic view

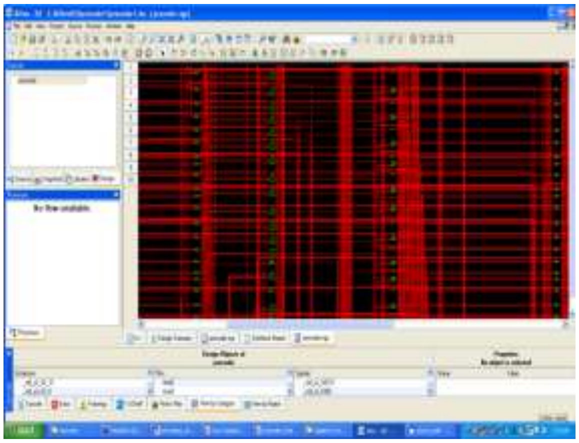

Fig 5.Technology Schematic view

V.PERFORMANCE ANALYSIS

This section addresses the design parameter selection for the fixed-complexity algorithms to be implemented in hardware. Additionally, the impact of applying an approximate norm for the computation of the distance increments is studied from an error-rate-performance point of view.

HARDWARE PARAMETERS	ESTIMATED RESULT
Slices	607
FFs	64
LUTs	1,117
IOs	07
Latency	36.166ns
Adders/Subtractors	30
Comparators	21
Gate counts	11,027
Power consumption	203mW

Table.1 Hardware Resource occupation

The above table clearly gives the resource utilization of our proposed system in terms of area requirement and power consumption. The Latency is also much reduced. Our main intension is to reduce the complexity and which leads to increase in efficiency.

Conclusion

This paper has addressed the issues of a fully-pipelined implementation of the FSE and K-best tree-search approaches for a 4 × 4 VP system. The sorting stages required by the K best scheme have been performed by means of the WPE distributed sorting strategy along with a non sequential complex plane enumerator, which has also been incorporated into the FSE structure to determine the child nodes to be expanded in those tree levels $i < N$ where $n_i > 1$. The design parameters that establish the performance complexity tradeoff of these non recursive tree-search approaches have been set so as to yield a similar count of allocated embedded multipliers. Additionally, the use of an approximate norm to Diminish the computational complexity of the PED calculations has been contemplated.

Due to the better performance, occupation results, and simplicity of implementation, it is concluded that the FSE is best suited for the practical implementation in terms of slices, flip flops, latency, comparators, gate counts and power consumption.

References

[1] *B. M. Hochwald, C. B. Peel, and A. L. Swindlehurst, "A vectorperturbation technique for near-capacity multiantenna multiuser communication, part II: perturbation," IEEE Transactions on Communications, vol. 53, no. 3, pp. 537–544, 2005.*

[2] *C. Windpassinger, R. F. H. Fischer, and J. B. Huber, "Latticereduction- aided broadcast precoding," IEEE Transactions on Communications, vol. 52, no. 12, pp. 2057–2060, 2004.*

[3] *A. K. Lenstra, H. W. Lenstra, and L. Lov´asz, "Factoring polynomials with rational coefficients," Mathematische Annalen, vol. 261, no. 4, pp. 515–534, 1982.*

[4] *L. Babai, "On lov´asz' lattice reduction and the nearest lattice point problem," Combinatorica, vol. 6, no. 1, pp. 1–13, 1986.*

[5] *D. Seethaler and G. Matz, "Efficient vector perturbation in multi-antenna multi-user systems based on approximate integer relations," in Proceedings of the EURASIP European Signal Processing Conference (EUSIPCO '06), pp. 1–5, September 2006.*

[6] *M. Taherzadeh, A. Mobasher, and A. K. Khandani, "Communication over MIMO broadcast channels using lattice-basis reduction," IEEE Transactions on Information Theory, vol. 53, no. 12, pp. 4567–4582, 2007.*

[7] *K. H. Lin, H. L. Lin, R. C. Chang, and C. F. Wu, "Hardware architecture of improved Tomlinson-Harashima Precoding for downlink MC-CDMA," in Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS '06), pp. 1200– 1203, December 2006.*

[8] *K.W.Wong, C. Y. Tsui,R. S.K.Cheng, andW. H.Mow, "AVLSI architecture of a k-best lattice decoding algorithm for MIMO channels," in Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '02), vol. 3, pp. 273–276, May 2002.*

[9] *L. G. Barberoand J. S.Thompson, "Rapidprototypingof a fixedthroughput sphere decoder for MIMO systems," in Proceedings of the IEEE International Conference on Communications (ICC '06), vol. 7, pp. 3082–3087, June 2006.*

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

[10]L. G. Barbero and J. S.Thompson, "FPGA design considerations in the implementation of a fixed-throughput sphere decoder for MIMO systems," in Proceedings of the International Conference on Field Programmable Logic andApplications (FPL '00), pp. 1–6, August 2006.

[11]L. G. Barbero and J. S. Thompson, "Extending a fixedcomplexity sphere decoder to obtain likelihood information.

[12]C. A. Shen and A.M. Eltawil, "A radius adaptive k-best decoder with early termination: algorithm and VLSI architecture," IEEE Transactions on Circuits and Systems I, vol. 57, no. 9, pp. 2476–2486, 2010.

[13]M. Mahdavi, M. Shabany, and B. V. Vahdat, "A modified complex k-best scheme for high-speed hard-output MIMO detectors," inProceedings of the 53rd IEEEInternationalMidwest Symposium on Circuits and Systems (MWSCAS '10), pp. 845– 848, August 2010

[14]S. Chen, T. Zhang, and Y. Xin, "Relaxed k-best MIMO signal detector design and VLSI implementation," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 15, no. 3, pp. 328–337, 2007.

[15]C. A. Shen, A. M. Eltawil, and K. N. Salama, "Evaluation framework for k-best sphere decoders," Journal of Circuits, Systems and Computers, vol. 19, no. 5, pp. 975–995, 2010