

# A VLSI Decoder for the Golden code

Barbara Cerato, Guido Masera, Emanuele Viterbo  
Dipartimento di Elettronica

Politecnico di Torino - TORINO (ITALY)

Email: {barbara.cerato, guido.masera, emanuele.viterbo}@polito.it

**Abstract**—The recently proposed Golden code is an optimal space-time block for 2 2 Multiple Input Multiple Output (MIMO) system, to be used in wireless transmission to achieve high reliability and throughput.

The aim of this work is the design of a VLSI decoder for a MIMO system coded with the Golden code. This architecture is based on the Sphere Decoding algorithm that allows Maximum Likelihood decoding performance, with polynomial search complexity. The proposed architecture exhibits an inherent flexibility with respect to different modulation schemes (4, 16, 64-QAM), which makes it particularly suitable for adaptive modulation schemes.

## I. INTRODUCTION

Designing very high data rate wireless channels with improved quality of service is one of nowadays greatest engineering and research challenges as proved by the increasing number of standards to regulate Wireless Local Area Networks (LAN) and Metropolitan Area Networks (MAN) (e.g. 802.11 and 802.16). In particular research is focused on two main objectives: high data rate and improved reliability.

Multiple Input Multiple Output (MIMO) systems by means of transmit antenna diversity combined with space-time coding are able to reach both these results [1]–[5].

This work is focused in the decoding of a  $2 \times 2$  MIMO signal coded with Golden code [3], which is a recently proposed code, chosen for being a full-rate and full-diversity code with optimal energy properties. A *Sphere Decoder* with a new simple architecture has been implemented to decode the Golden code.

### A. Golden code

The Golden code is a Space-Time code for a  $2 \times 2$  coherent MIMO channel; it was found independently by [3]–[5]. It is one of the new full-rate and full-diversity codes able to achieve the best trade-off between data rate and diversity gain. Differently from previously found codes, that suffer from a loss of spectral efficiency with the increase of the signal constellation, it can be used also with higher order modulations. For this reason it can be successfully employed in systems with adaptive selection of the modulation.

The codewords  $\mathbf{X}$  of the Golden code are  $2 \times 2$  complex matrices of the following form :

$$\mathbf{X} = \frac{1}{\sqrt{5}} \begin{bmatrix} \alpha[a + b\theta] & \alpha[c + d\theta] \\ i\sigma(\alpha)[c + c\sigma(\theta)] & \sigma(\alpha)[a + b\sigma(\theta)] \end{bmatrix} \quad (1)$$

where  $a, b, c, d$  are the information symbols chosen in a  $Q^2$ -QAM constellation,  $i = \sqrt{-1}$ ,  $\theta = (1 + \sqrt{5})/2 = 1.618\dots$  (Golden number),  $\sigma(\theta) = 1 - \theta$ ,  $\alpha = 1 + i$ ,  $\sigma(\alpha) = 1 + i\theta$  [6].

### B. The $2 \times 2$ MIMO System Model

Golden code is applied to 2-transmit 2-receive antenna MIMO systems and requires two channel uses to transmit a codeword. In order to model the  $2 \times 2$  MIMO channel, its impulse response can be used. Assuming  $h_{ij}$  as the time-varying channel fading coefficients between the  $j$ -th transmit antenna and the  $i$ -th receive antenna, the MIMO channel is described through a  $2 \times 2$  matrix:

$$\mathcal{H} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \quad (2)$$

where  $h_{ij} \sim \mathcal{N}_c(0, 1)$ . Assuming the “Block Fading” channel model, each transmitted codeword will be affected by an independently varying channel matrix  $\mathcal{H}$ . Then, the  $2 \times 2$  received matrix is

$$\mathbf{Y} = \mathcal{H}\mathbf{X} + \mathbf{Z}$$

where  $\mathbf{Z}$  is the additive white Gaussian noise matrix with entries  $\sim \mathcal{N}_c(0, N_0)$ .

The codeword  $\mathbf{X}$  can be represented in vectorized form where, furthermore, real and imaginary components are separated, resulting in a  $8 \times 1$  real vector  $\mathbf{x}$ . It can be seen that  $\mathbf{x} = \mathbf{B}\mathbf{s}$ , where  $\mathbf{B}$  is a  $8 \times 8$  orthogonal matrix ( $\mathbf{B}^{-1} = \mathbf{B}^T$ ) and  $\mathbf{s} = (\Re a, \Im a, \Re b, \Im b, \Re c, \Im c, \Re d, \Im d)$ , [6].

Channel matrix  $\mathcal{H}$  consequently can be rearranged in a  $8 \times 8$  real-valued matrix  $\mathbf{H}$ .

The Golden coded system model can so be expressed as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z} \quad (3)$$

where  $\mathbf{y}$  is the  $8 \times 1$  received real vector and  $\mathbf{z}$  is a 8-dimensional i.i.d. zero mean Gaussian noise real vector.

Decoding the Golden code is equivalent to decoding an 8-dimensional lattice with generator matrix  $\mathbf{M} = \mathbf{H}\mathbf{B}$ . Provided that  $\mathbf{H}$  is perfectly known at the receiver, the optimal detector for a MIMO channel, which minimizes the codeword error rate, is the maximum likelihood (ML) detector that solves the following equation:

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s} \in Q^n} \|\mathbf{y} - \mathbf{M}\mathbf{s}\|^2 \quad (4)$$

where  $Q^n$  is the cardinality of the search space and  $n = 8$ .

The above expression represents a least-square (LS) minimization problem. Exhaustive search of the ML solution

has exponential complexity and in this particular case it has  $2^{n \log_2 Q}$  possible solutions. Sphere decoding algorithms have then been proposed with the purpose of decreasing the decoder complexity.

## II. SPHERE DECODING ALGORITHM

*Sphere Decoding Algorithms* [7] aims at finding the ML solution by analyzing only a subset of the solution space. While minimization (4) for an arbitrary  $M$  and  $\mathbf{y}$  is NP-hard, it has been shown [8] that, in a certain range of parameters, which is not too far from those of real systems, the average complexity of sphere decoder is polynomial.

A hypersphere is constructed around the received vector and only points inside it are taken into account. since the others are actually too far. This constraint can be written as:

$$\|\mathbf{y} - M\mathbf{s}\|^2 \leq C_0 \quad (5)$$

where  $C_0$  is the radius of the hypersphere. This algorithm can be viewed compound by two phases.

1) *Tree construction*: With a linear transformation of the  $M$  matrix, such as QR or Cholesky decomposition, it is possible to rewrite it as a product of two matrices, one of which upper triangular [7]. In this work, QR decomposition has been employed so that, imposing  $M = QR$ , (4) can be rewritten as:

$$\begin{aligned} \arg \min_{\mathbf{s} \in Q^n} \|\mathbf{y} - QR\mathbf{s}\|^2 &= \arg \min_{\mathbf{s} \in Q^n} \|\mathbf{Q}^T \mathbf{y} - R\mathbf{s}\|^2 \\ &= \arg \min_{\mathbf{s} \in Q^n} \|\tilde{\mathbf{y}} - R\mathbf{s}\|^2 \end{aligned} \quad (6)$$

where we have exploited the orthogonality of  $Q$  and  $\tilde{\mathbf{y}} = \mathbf{Q}^T \mathbf{y}$  represents the “zero-forcing” (ZF) solution. The upper triangular structure of the factored matrix enables to take every component separately into account for the computation of the distance between the two points. This distance  $d^2(\mathbf{s}) = \|\tilde{\mathbf{y}} - R\mathbf{s}\|^2$  can so be decomposed recursively as:

$$T_l(\mathbf{s}^{(l)}) = \begin{cases} 0 & \text{if } l = n + 1 \\ T_{l+1}(\mathbf{s}^{(l+1)}) + |\tilde{y}_l - \sum_{j=l}^n R_{lj}s_j|^2 \\ = T_{l+1}(\mathbf{s}^{(l+1)}) + |\tilde{y}_l - \sum_{j=l+1}^n R_{lj}s_j - R_{ll}s_l|^2 \\ = T_{l+1}(\mathbf{s}^{(l+1)}) + |\psi_{l+1} - R_{ll}s_l|^2 & \text{otherwise} \end{cases} \quad (7)$$

where  $\mathbf{s}^{(l)} = [s_l \ s_{l+1} \ \dots \ s_n]$ ,  $T_1(\mathbf{s}) = d^2(\mathbf{s})$  and  $\psi_{l+1} = \tilde{y}_l - \sum_{j=l+1}^n R_{lj}s_j$ , in particular  $\psi_{N+1} = \tilde{y}_l$ .

One of the most interesting consequences of this interpretation is that the exploration of the lattice can be thought as a tree traversal. This tree has  $N$  levels and every node at each level has  $Q$  sons. At every level the radius constraint (5) must be verified and satisfied, otherwise the branch is pruned. Figure 1 depicts a two level tree for a QPSK modulation.  $T_l$  is the distance metric at level  $l$  in (7). It can be noticed that at the deepest level  $\psi_3$  has been substituted by its initial value  $\tilde{y}_2$ . At

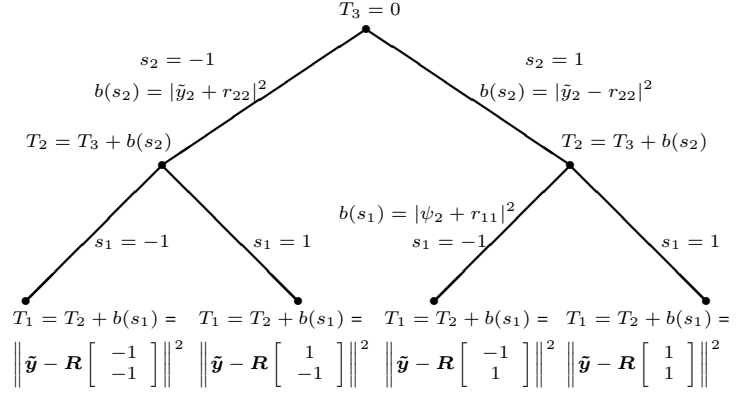


Fig. 1. Two level set partitioning of  $\Lambda$ , output label  $s_2$  selects the first level and  $s_1$  selects the second level in the partition tree.

the lowest level final metrics are explicitly calculated for this simple case.

2) *Tree exploration*: Several algorithms have been studied in order to make the tree traversal efficient. First algorithm, proposed by Pohst in [9] needs to choose explicitly an initial radius. This is a very critical choice: if a too large radius is chosen, too many points fall into the hypersphere, while for a too small radius no points are left inside it. A more efficient algorithm, which does not need the choice of the initial radius, has been proposed by Schnorr and Euchner (SE) [10]. An infinite initial radius is selected and, following the direction of the zero-forcing solution  $\tilde{\mathbf{y}}$ , a depth-first traversal of the tree is performed. Radius is progressively reduced every time a leaf is reached and distance between two points is smaller than current radius. Originally thought for infinite lattices, this algorithm has been adapted to finite lattices [7].

## III. HARDWARE IMPLEMENTATION

The decoding phase of a Golden code MIMO system is mainly constituted by three blocks: QR decomposition, column reordering and tree search. While column reordering optionally reduces the tree-search complexity, QR decomposition is mandatory because it allows constructing the tree and finding the ZF solution of the LS minimization problem.

These two blocks perform preprocessing tasks, working on the lattice generator matrix  $M = HB$ ; since the code generator matrix  $B$  is constant, they require to be applied at the channel estimation update frequency. The refresh frequency of the channel estimation can terribly change, from one application to another, in order to fit the standard purposes and scenarios, but it is generally almost one or two orders of magnitude lower than the signal rate. For this reason the column reordering and the QR decomposition blocks are not the critical components in the decoder architecture.

Main focus of this work is on the hardware design of the tree-search algorithm for the detection of the Golden code, which is the most computational intensive block.

Two are the main implementation requirements taken into account in the design of the proposed architecture. On one

hand, the decoder has been conceived with the purpose of supporting a certain degree of flexibility with respect to the choice of the modulation scheme and to the size of the search space. On the other hand, the achievement of high decoding throughputs has been taken as the second basic design objective.

The block scheme of the SE tree-traversal circuit is depicted in Figure 2. Three fundamental blocks can be identified in the shown architecture:

- **U\_psi unit**, which selects the most probable son of the current node and computes updated  $\psi_l$
- **U\_psi\_step unit**, which selects an alternative node to be expanded and computes for this node the same amount
- **Metric\_compute**, which computes metric of the current node  $T_l$  (7).

The core of the traversal algorithm is constituted by another essential block, represented in Figure 2 as **C.U.** (Control Unit) and devoted to properly select the correct tree search direction. Moreover it has also two additional tasks: to verify pruning conditions and to properly dispatch data between involved units, on the basis of the performed verification.

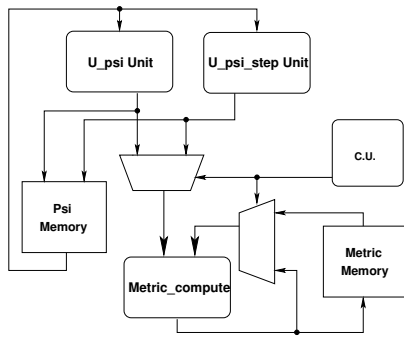


Fig. 2. Sphere Decoder block scheme.

Comparing this architecture to other implementations [11], it can be observed that a single metric computation is performed at each cycle, instead of multiple parallel metric computations. This characteristic justifies the reduced complexity that will be shown in Section IV and it also enables the inherent flexibility of the proposed architecture.

The direct implementation of eq. (7), that is the computation of the  $\psi_l - R_{ll}s_l$  value for all possible values of  $s_l$ , in order to select the best son for the considered node, tends to become infeasible when the order of the modulation increases, due to the large number of required operations. In order to avoid this limitation, the computation in (7) was rearranged in two consecutive steps. In the first step, **U\_psi** Unit exploits the knowledge of the previous value of  $\psi_l$  to compute the division  $\psi_l/R_{ll}$  and to retrieve the value of  $s_l$  for the analyzed “father node”. This allows obtaining the updated  $\psi_{l+1}$  values for the candidate sons (Fig. 3) and selecting the one with the minimum value. At the second step, the actual metric value,  $T_l$ , is computed for the selected node (**Metric\_compute** unit).

The need for multiple metric computation is avoided by means of this two-step approach. This solution achieves low complexity and, at the same time, flexibility in terms of supported modulation schemes.

It is worth noting that we do not need to compute the exact solution of division  $\psi_l/R_{ll}$ , since we are only interested in some specific values of the signal  $s_l$ , i.e. the integer values ( $-3, -1, 1, 3, \dots$ ). Thus, we find the closest integer solution of the division and if the approximation is by defect or by excess: this computation is easily performed, as it only requires shifts and subtractions.

The addressed requirements of reduced complexity and flexibility cannot be achieved at the expenses of processing throughput. In previous published approaches, high throughput is obtained resorting to parallel architectures and two different kinds of parallelism are usually employed:

- Parallelism at the level of tree exploration
- Parallelism at the level of the metric computation for all sons of a given node and in the selection of the most probable son.

The first technique can be used only with some suboptimal algorithms [12] and it becomes unfeasible when optimal algorithms are adopted, since it requires large amounts of hardware resources. The second approach is feasible only in small QAM modulation schemes as it implies many concurrent multiplications. Thus these techniques are not viable for the implementation of flexible architectures. As a consequence, in this work, parallelism relies on a *pipelining-like* approach.

During tree exploration, while the metrics of a given father node are evaluated by the **Metric-Compute Unit**, two “candidate” nodes are concurrently computed: the first one is a direct son of the considered node (if the father node is a leaf we avoid this evaluation), while the second one is an “alternative” node. This alternative node may be at the same level as the father node or even at a deeper level in the tree. While the first elaboration is performed by the **U\_psi** unit, the alternative node computation is concurrently performed by the **U\_psi\_step** sub-circuit (see Figure 2) and both of them compute novel  $\psi_{l+1}$  values for the next step in the tree traversal. Depending on the values assumed by the father node metric, the algorithm descends along the tree, reaching the son node, or it moves to the alternative node.

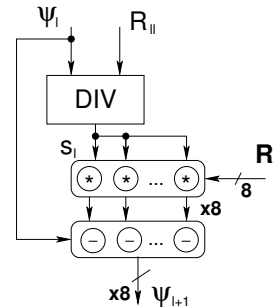


Fig. 3. U\_psi Unit datapath

TABLE I  
SYNTHESIS RESULTS

Reference	[11]	[13]	This work	
Antennas	4×4	4×4	2×2 per two channel uses	
Modulation	16-QAM	16-QAM	4,16,64-QAM	16-QAM
Detector	depth-first sphere	K-best sphere	depth-first sphere	
BER Perf.	ML	Quasi-ML	ML	
Tech. [ $\mu\text{m}$ ]	0.25	0.35	0.13	0.25
Core Area [GE]	117k	91k	68k	56k
	+preproc.	+preproc.	+preproc.	+preproc.
Max. Clock	51 MHz	100 MHz	208 MHz	109 MHz
Throughput @SNR=20 dB	73 Mbps	52 Mbps	145.1 Mbps [16-QAM]	83 Mbps

#### IV. RESULTS

The proposed architecture has been extensively simulated. System performance are reported in Figure 4 in terms of bit error rate(BER) vs. SNR, for 4, 16 and 64-QAM modulations. Fixed-point results are also plotted for the same modulation schemes: finite precision data have been derived adopting a 16 bits datapath, with 7 bits devoted to the mantissa and 9 bits for the fractional part. In the plots no performance loss can be appreciated with respect to floating-point simulations, here introduced for comparison purposes. We note that the 16 bits precision is enough also for the most critical 64 QAM modulation.

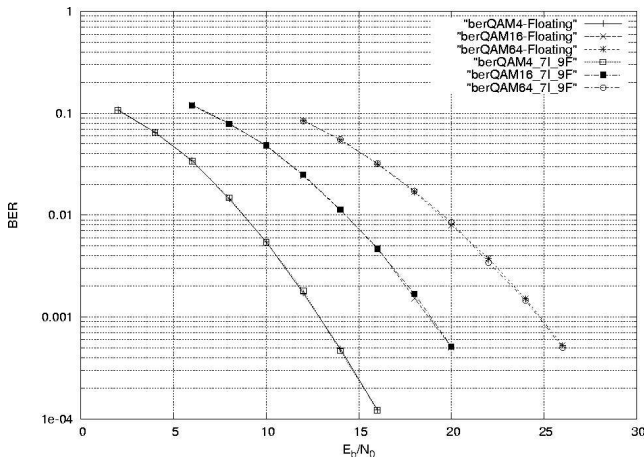


Fig. 4. Proposed system performance with different modulations.

The proposed architecture has then been synthesized on both 0.13 $\mu\text{m}$  and 0.25 $\mu\text{m}$  CMOS Standard Cell technologies.

In order to enable the direct comparison with existing hardware realizations [11], [13], a 16 bits datapath has been chosen and the overall decoder has also been simulated with the V\_BLAST 4×4 MIMO system and throughput figures reported in table I refer to this configuration.

Analyzing data in Table I, it can be noticed that our rearranged approach with a single metric computation per cycle, allows a significant complexity reduction (approx. 50% for 16

QAM modulation) with respect to parallel structures. At the same time, thanks to the adopted pipelined architecture, we can achieve remarkable average decoding throughput without any highly specialized structure. Moreover, our flexible decoder is not limited to a single modulation scheme, but it can adapt to different modulations (4, 16 and 64-QAM).

#### V. CONCLUSION

A novel Golden code MIMO decoder based on a rearranged SE Sphere Decoding algorithm has been proposed. It achieves an average throughput of 83 Mbps on a 0.25  $\mu\text{m}$  Standard Cell CMOS technology for a 16-QAM modulation and an overall complexity limited to 56 kGates. The flexible version of the designed architecture is able to sustain multiple modulation schemes and requires 68 kGates, achieving an average throughput of 145 Mbps on a commercial 0.13  $\mu\text{m}$  Standard Cell technology.

#### ACKNOWLEDGMENT

This work was supported by the STREP project No. IST-026905 (MASCOT) within the Sixth Framework Programme of the European Commission.

#### REFERENCES

- [1] B. Hassibi and B. Hochwald, "High-rate codes that are linear in space and time," *IEEE Trans. Inform. Theory*, vol. 48, no. 7, pp. 1804–1824, July 2002.
- [2] B. A. Sethuraman, B. S. Rajan, and V. Shashidhar, "Full-diversity, highrate space-time block codes from division algebras," *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2596–2616, Oct. 2003.
- [3] J.-C. Belfiore, G. Rekaya, and E. Viterbo, "The Golden code: A 2 x 2 full-rate space-time code with non-vanishing determinants," *IEEE Trans. Inform. Theory*, vol. 51, no. 4, pp. 1432–1436, April 2005.
- [4] H. Yao and G. Wornell, "Achieving the full MIMO diversity-multiplexing frontier with rotation-based space-time codes," in *Proc. of Allerton Conf. on Communication, Control and Computing*, October 2003.
- [5] P. Dayal and M. Varanasi, "An optimal two transmit antenna space-time code and its stacked extensions," in *Proc. of Asilomar Conf. on Signals, Systems and Computers*, November 2003.
- [6] E. Viterbo. (2006) The Golden Code Homepage. [Online]. Available: [http://www1.tlc.polito.it/~viterbo/perfect\\_codes/Golden\\_Code.html](http://www1.tlc.polito.it/~viterbo/perfect_codes/Golden_Code.html)
- [7] M. O. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search of the closest lattice point," *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2389–2402, October 2003.
- [8] H. Hassibi and B. Vikalo, "The expected complexity of sphere decoding, Part I: Theory, Part II: Applications," *IEEE Trans. Signal Processing*, vol. 44, no. 170, pp. 463–471, April 1985.
- [9] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Computat.*, vol. 44, no. 170, pp. 463–471, April 1985.
- [10] C. Schnorr and M. Euchner, "Lattice basis reduction: improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, no. 2, pp. 181–191, September 1994.
- [11] A. Burg, M. Borgmann, M. Wenk, M. Zellwegger, W. Fichtner, and H. Bölcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, July 2005.
- [12] K. Wong, C. Tsui, R. S. Cheng, and W. Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," in *Proc. IEEE ISCAS'02*, 2002, pp. 273–276.
- [13] Z. Guo and P. Nilsson, "A VLSI architecture for the Schnorr-Euchner decoder for MIMO systems," in *Proc. IEEE CAS Symp. Emerging Technologies*, June 2004, pp. 65–68.