Low Complexity QR-Decomposition Architecture using the Logarithmic Number System

Jochen Rust, Frank Ludwig, Steffen Paul
Institute of Electrodynamics and Microelectronics (ITEM.me)
University of Bremen, Bremen, Germany, +49(0)421/218-62538
Email: {rust, ludwig, steffen.paul}@me.uni-bremen.de

***Abstract*—In this paper we propose a QR-decomposition hardware implementation that processes complex calculations in the logarithmic number system. Thus, low complexity numeric format converters are installed, using nonuniform piecewise and multiplier-less function approximation. The proposed algorithm is simulated with several different configurations in a downlink precoding environment for 4×4 and 8×8 multi-antenna wireless communication systems. In addition, the results are compared to default CORDIC-based architectures.**

In a second step, HDL implementation as well as logical and physical CMOS synthesis are performed. The comparison to actual references highlight our approach as highly efficient in terms of hardware complexity and accuracy.

***Index Terms*—QR-Decomposition, Nonuniform function approximation, LNS**

I. Introduction

The QR-decomposition (QRD) is an essential algorithm in order to solve efficiently a system of linear equations. In the context of wireless communication, the QRD has a significant impact on several tasks. Especially in multi-antenna systems, the QRD of the channel matrix is a prerequisite e.g. for the symbol detection at the receiver side [1]. Another important application is the precoding, at which the channel equalization task is transferred to the transmitter in order to reduce the complexity and the power usage of the receivers. For this transmitter-side pre-equalization, different approaches are known. This contribution comprises QRD processing for a Tomlinson-Harashima precoding (THP) in a multi-antenna system [2].

In this paper, a QRD algorithm is introduced that is based on calculations in the logarithmic number system (LNS) [3]. Its major advantage refers to the simplification of several algebraic operations and elementary functions by processing the corresponding exponential concatenations. Logarithmic (LOG) and anti-logarithmic (ALOG) converters are used, enabling the number system transformation between LNS and fixed-point number system (FPNS). As this requires a complex function calculation, it is realized by regression-based interpolation. In order to minimize processing time and hardware complexity, nonuniform and multiplier-less piecewise function approximation (NPA) is used [4]. Several NPA-based LOG and ALOG converters with different approximation accuracies are investigated, as this will directly affect both timing and chip area of a corresponding hardware implementation.

The work presented in this article is partly funded by the German Research Foundation (DFG) under grant PA 438/3-2 and PA 438/6-1.

978-3-9815370-0-0/DATE13/ © 2013 EDAA

$$\mathbf{H} = \begin{bmatrix} \mathbb{C} & \mathbb{C} & \mathbb{C} \\ \mathbb{C} & \mathbb{C} & \mathbb{C} \\ \mathbb{C} & \mathbb{C} & \mathbb{C} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbb{R} & \mathbb{C} & \mathbb{C} \\ \mathbb{R} & \mathbb{C} & \mathbb{C} \\ \mathbb{R} & \mathbb{C} & \mathbb{C} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbb{R} & \mathbb{C} & \mathbb{C} \\ 0 & \mathbb{C} & \mathbb{C} \\ 0 & \mathbb{C} & \mathbb{C} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbb{R} & \mathbb{C} & \mathbb{C} \\ 0 & \mathbb{R} & \mathbb{C} \\ 0 & \mathbb{R} & \mathbb{C} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbb{R} & \mathbb{C} & \mathbb{C} \\ 0 & \mathbb{R} & \mathbb{C} \\ 0 & 0 & \mathbb{C} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbb{R} & \mathbb{C} & \mathbb{C} \\ 0 & \mathbb{R} & \mathbb{C} \\ 0 & 0 & \mathbb{R} \end{bmatrix}$$

Fig. 1. Example of the proposed QR-Decomposition processing applied to a 3×3 complex valued matrix. In a first step the imaginary values are rotated to zero and applied both to $\mathbf{H}$ and the unitary matrix $\mathbf{Q}$. Next, this is applied to the real values.

The following two Sections provide an overview about most common QRD algorithms as well as the LNS and its converters. Section IV is primarily concerned with the hardware architecture. Next, the results of simulation and CMOS design are presented before the most important aspects are summarized in the last Section.

II. QR decomposition

Concerning hardware efficient QRD design, there are in general three different methods that must be distinguished: the Householder transformation, the Gram-Schmidt process and Givens rotation (sometimes also called Jacobi rotation) [5]. For a hardware implementation, the Givens rotation is mostly favored, as the Householder transformation cannot be parallelized and the Gram-Schmidt algorithm is numerically unstable. Furthermore, the Gram-Schmidt process requires square-root operations as well as the numerical more stable modified Gram-Schmidt [5]. On the other hand, the Givens rotation allows a parallel computational structure. Also the well-known CORDIC algorithm [6] can be used to avoid crucial algebraic calculations like division or inverse computation. Besides, CORDIC-based approaches require only low hardware complexity, as they are based on iterative result refinement by accumulating microrotation steps.

Fig. 1 briefly demonstrates the processing steps to decompose the complex 3×3 matrix $\mathbf{H}$ in the right triangular matrix $\mathbf{R}$. First, the imaginary parts of the first column are eliminated

with three Givens rotations (one rotation per column). With two further Givens rotations the real parts of the second and third elements of the first column are eliminated. As shown in Fig. 1, this procedure will be repeated several times such that the matrix finally results in a right triangular matrix $\mathbf{R}$ with real-valued main diagonal elements. All executed Givens rotations on $\mathbf{H}$ are additionally applied to the unitary matrix $\mathbf{Q}$, which is initialized with an identity matrix.

The Givens rotation itself is performed by the multiplication of the Givens matrix

$$\mathbf{G} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \tag{1}$$

with the corresponding rows of $\mathbf{R}$ and $\mathbf{Q}$. c and s denote the (co-)sine values that are calculated by

$$c = \frac{x}{\sqrt{x^2 + y^2}} \quad \text{and} \tag{2}$$

$$s = -\frac{y}{\sqrt{x^2 + y^2}} \, . \tag{3}$$

A more detailed description of the QRD and the Givens rotation can be found in [5].

III. Logarithmic number system

As described previously, LNS simplifies the calculation effort of several algebraic operations and elementary functions by processing the corresponding exponential concatenation. However, some elementary operations like addition and subtraction cannot be performed in LNS by trivial means. Thus, numeric format transformation is a popular solution for this issue, even though this requires complex function calculation. In the scope of digital signal processing, a fundamental approach for correspondent converter modules has been proposed by Mitchell [7] in 1962. His algorithm is based on a single straight linear approximation $\log_2(x+1) \approx x$ and $2^x - 1 \approx x$ for LOG and ALOG, which allows number system transformation by shift-based value normalizing, but achieves bad accuracy in many cases. Thus, minimizing the approximation error is a major concern to improve this idea.

A. Numeral system

In contrast to FPNS that depicts a value N by a sign S, a value offset O and a radix point r

$$N = (-1)^S \cdot O \cdot 2^{-r} \, , \tag{4}$$

LNS represents a value by the sign bit S, a zero bit Z, an exponent E and a logarithmic mantissa L. A popular numeral representation is given by Detrey et al. [3] for LNS, where the fractions are concatenated similar to default floating-point numeral systems. Thus, the numeral system we use in this paper specifies a value N by

$$N = (1 - Z) \cdot (-1)^S \cdot 2^{-E} \cdot 2^L \, . \tag{5}$$

In order to simplify the (anti-)logarithmic conversion, normalized logarithmic mantissa and fixed-point offset values are used as described above. Thus, the transformations can be expressed as

$$L = \log_2(O_n + 1); \; O_n = O \cdot 2^m; \; 0 \leq O_n < 1 \tag{6}$$

and

$$O = 1 + (2^{L_n} - 1); \; L_n = L \cdot m; \; 0 \leq L_n < 1 \, , \tag{7}$$

with O_n and L_n as the normalized values and m as the shift coefficient. Both modifications keep the resulting values in suitable ranges. However, additional computation effort is necessary in order to determine L_n and O_n. Because both values are modified by 2^m multiplications, this is easily designable in hardware by shift operators. The increase by one can be simply considered inside the shift block of the ALOG unit. The entire structure of LOG and ALOG is given in Fig 2.

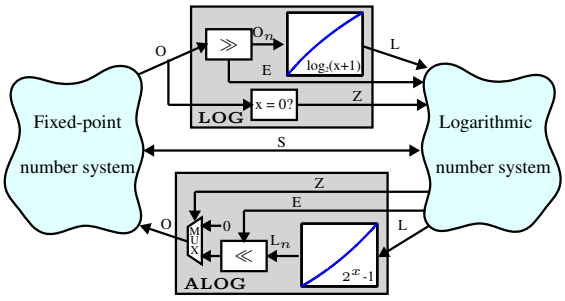

Fig. 2. Architecture of LOG and ALOG converter for number system transformation.

B. Converter design

For efficient computation of the converter functions $\log_2(x+1)$ and $2^x - 1$, we propose NPA-based hardware implementation, as it has proven to be a powerful solution for function approximation [4]. Its main idea refers to linear regression based function course emulation, reducing computation time and latency of elementary functions at the cost of accuracy [8]. In general, linear equations are expressible as

$$\tilde{f}(x) = \alpha_0 x + \beta_0 \quad , \tag{8}$$

where x denotes the input data, $\tilde{f}(x)$ the approximation of the original function $f(x)$ and α_0 and β_0 the gradient and offset, respectively. For NPA, the original function is split up into several sub-functions of variable input range each (see Fig. 3). In order to enable quick access to each sub-function, only a restricted segmentation of the original function is allowed. Thus, each sub-function segment must fulfill

$$\mathrm{seg}(k) = \mathrm{seg}(k-1) + \frac{\mathrm{B} - \mathrm{A}}{2^{h_k}}, \tag{9}$$

with A, B as start and end point of the function, seg as sub-function start point, k as segment index ($\mathrm{seg}(0) = \mathrm{A}$) and $h_k \in \mathbb{N}^+$ as the interval exponent of the kth segment. In addition, the input range of the original function is set to

$$\mathrm{B} - \mathrm{A} = 2^{h_{max}}. \tag{10}$$

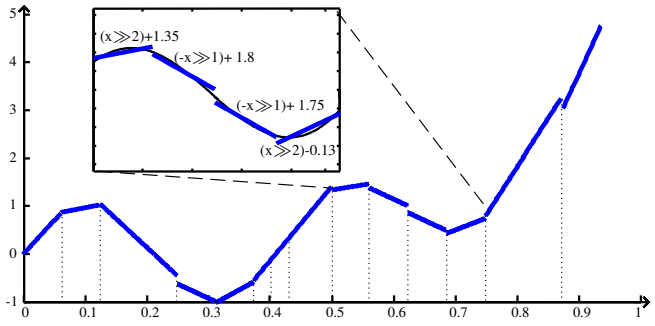

Fig. 3. Example of nonuniform multiplier-less piecewise function approximation.

These constraints enable the selection of each segment only by taking the most significant bits (MSB) of x into account. Note, that h_k may differ for each segment which may cause a varying number of MSB that must be considered.

For further calculation decrease, multiplier-less linear equations are used in this scope. In detail, only a limited set of partial products of a common tree-multiplier is regarded here [9]. This reduces the gradient calculation effort to shift and add operations, the total number of which is specified by the quantization factor (QF). Thus, NPA can be expressed

$$\tilde{f}(\mathbf{x}) = \left[\sum_{j=0}^{l-1} \pm \begin{pmatrix} 2^{\lambda_{0,j}} \\ 2^{\lambda_{1,j}} \\ \dots \\ 2^{\lambda_{k-1,j}} \end{pmatrix} \mathbf{x} + \boldsymbol{\beta} \right] \cdot \boldsymbol{\kappa}(x) \qquad (11)$$

with $\lambda_{i,j}$ as the exponent of the actual partial product, $\boldsymbol{\beta}$ as vector of offsets and k, l determining the total number of segments and partial products, respectively. The $\boldsymbol{\kappa}$ vector is responsible for the selection of the actual valid linear equation

$$\boldsymbol{\kappa}(x) = \begin{cases} (1,0,...,0)^T; & \mathrm{A} \le x < \mathrm{seg}(1) \\ (0,1,...,0)^T; & \mathrm{seg}(1) \le x < \mathrm{seg}(2) \\ \vdots & \\ (0,0,...,1)^T; & \mathrm{seg}(k-1) \le x < \mathrm{B} \end{cases} . \qquad (12)$$

A graphical example of the NPA-based approximation scheme is given in Fig. 3.

IV. Architecture

In order to achieve best results in terms of low complexity implementation, a hardwired hardware structure is chosen [10]. The arithmetic calculations are performed in FPNS and LNS, as this allows the simplification of all operations to additions, subtractions or shifts, but requires the previously described converter units. 16 and 18 bit sized data paths are used, as described in detail later on (see Sec.V-B). The architecture consists of a memory block as well as of a control and data path unit. As only a total number of 64 memory slots is required for complex 4×4 matrices, a register array implementation is chosen. Note, that only LNS-based values are stored in the memory as this is the default numeric format and improves the overall QRD calculation performance.

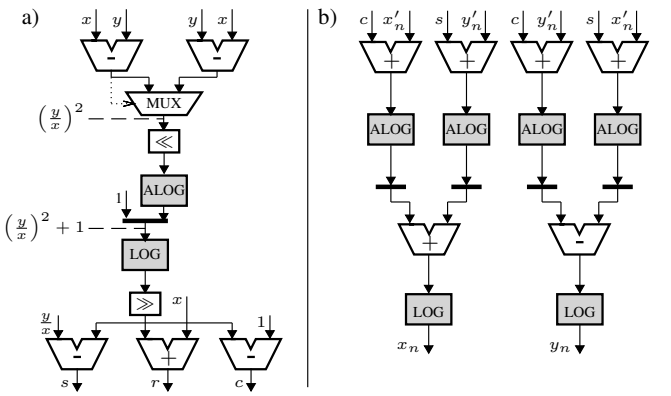

Fig. 4. Data path of LNS-based QRD with a) GR1 and b) GR2 and $x > y$. The solid, dotted and dashed lines depict the data flow, the control signal and interim results, respectively. x'_n and y'_n denote the unrotated inputs as well as x_n and y_n are the resulting outputs.

A. Control unit

The control unit possesses an input/output(IO)-port module, which is used to load and store external values to the memory. Further, a finite state machine (FSM) is located here handling the global processing of the QRD algorithm. Address decoding related to the current FSM state is performed, in the same way as described in Fig. 1. Also a control signal denoting the calculation of s, c and $r = \sqrt{x^2 + y^2}$ (GR1) or its application within Givens rotation processing (GR2) is generated here.

In order to keep the address decoding task as simple as possible, it is directly derivable from the current FSM state. Thus, the FSM is implemented as counter with a variable increment.

B. Data path

The data path processing can be separated into the GR1 and GR2 calculation mode. Both modes are realized by a straight forward implementation of the Givens-Rotation, similar to the algorithm mentioned in Sec. II. In order to keep the values in convenient ranges and to save one addition (see Fig. 4a), GR1 processing is transformed to

$$\begin{aligned} r &= x\sqrt{1+\tfrac{y}{x}^2} \;; x > y \;\vee\; r = y\sqrt{1+\tfrac{x}{y}^2} \;; y \ge x \\ c &= \frac{1}{\sqrt{1+\tfrac{y}{x}^2}} \;; x > y \;\vee\; c = \frac{\tfrac{x}{y}}{\sqrt{1+\tfrac{x}{y}^2}} \;; y \ge x \\ s &= \frac{\tfrac{y}{x}}{\sqrt{1+\tfrac{y}{x}^2}} \;; x > y \;\vee\; s = \frac{1}{\sqrt{1+\tfrac{x}{y}^2}} \;; y \ge x \end{aligned} . \qquad (13)$$

As mentioned above, the LNS is used as default numeric format. However, for simple additions and subtractions, which must also be considered, LOG and ALOG units are installed.

The GR1 calculation starts with calculating the quotients of the input operands. By regarding the exponent of the result, the smaller value can be selected. Next, squaring is performed by a single shift. For the addition, the value is converted to FPNS. Because of the introduced processing modifications, the incrementation effort simplifies to a single bit flip. The square-root operation is achieved by LOG conversion followed by a shift. Finally, the requested variables r, c and s are achieved by

TABLE I
APPROXIMATION RESULTS OF THE THREE DIFFERENT LNS CONFIGURATIONS COMPARED TO DEFAULT CORDIC PERFORMANCE.

Configuration	CORDIC†	LNS$_I$	LNS$_{II}$	LNS$_{III}$
Latency	m	4	4	4
Additions GR1	$2m$	7	7	7
Additions GR2	$2m$	12	12	12
Segments LOG	-	7	65	643
Segments ALOG	-	7	67	665

†*m denotes the number of microrotations*

multiplication and division that are implemented as subtracters and adders, respectively. Note, that the operand selection must be taken into account at this step.

For GR2 mode, the processing starts with the calculation of the four vector rotation summands by LNS-based addition. As these values must be added or subtracted for vector rotation, FPNS transformation is again required. In a last step, LOG conversion is performed in order to write back the results in LNS. An architectural overview of the data path is given in Fig. 4.

In order to achieve convenient results in terms of low complexity, time-sharing is applied [11]. Thus, several adder units inside the data path are merged. GR1 is processed in two cycles, inserting a pipeline stage after the ALOG conversion. Unused adders are disabled by operand isolation [12]. For GR2, the four summands are calculated first, transformed into FPNS and stored in pipeline registers, for which two cycles are required. In a third cycle, the values are summed and converted back into LNS. In order to speed-up QRD processing and to avoid divisions by zero, the Z flag is investigated before data path calculations are started. In GR1 mode, the entire row processing is skipped if y is zero. For GR2, no calculation is performed if both x and y are zero.

V. RESULTS

A. Automatic converter approximation

The converter approximation is performed automatically, considering diverse design constraints. Thus, the QF, an average accuracy and size of data path must be specified. Based on these parameters, a linear equation is defined and its approximation quality to the corresponding (sub-)function is estimated. In detail, the quantized gradient α as well as its offset β are estimated by comparing them to the default linear approximation. If the specified average accuracy constrained is violated, bisection is performed. Due to (10), both resulting sub-functions fulfill the segment equation (9). All remaining sub-sections are investigated straight forward from the lower border A to the upper border B. After the function approximation is completed for all sub-functions, the resulting linear equations are mapped to verilog code. In detail, parameters are extracted that are mapped automatically to corresponding hardware elements. So e.g. for segment selection, multiplexer units are used and the multiplier quantization is realized by adder and subtracter units.

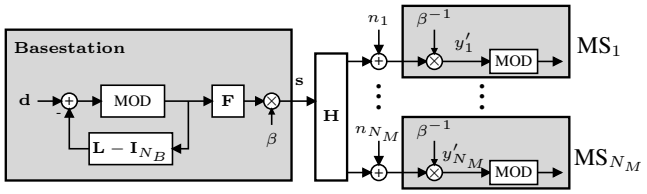

Fig. 5. THP structure in the base station for downlink transmission with decentralized mobile stations (MS).

For LOG and ALOG function approximation, QF is set to one. Moreover, three different configurations LNS$_I$, LNS$_{II}$ and LNS$_{III}$ are investigated that possess an equal exponent length ($E = 5$). The offset O and the mantissa L varies, due to the overall data path size. Thus, LNS$_I$ is realized with an overall data path size of 16 bit while LNS$_{II}$ and LNS$_{III}$ are realized with 18 bit, which refer to $L = 9$, Q3.12 and $L = 11$, Q3.14, respectively. In addition, approximation accuracies for the converter functions of 0.01, 0.001 and 0.0001 are chosen for LNS$_I$, LNS$_{II}$ and LNS$_{III}$, respectively. Note, that the accuracies are adjusted to the Q.15 format.

For delay evaluation, the longest sequence of traversed adders from data input to output is considered, as these modules have a major impact on the delay. Thus, each LNS configuration has a delay equal to a CORDIC-based data path with four iteration steps. In terms of overall adder complexity, GR1 and GR2 outperform CORDIC designs as they require only four and six iterations, respectively. An overview is given in Tab. I. As these results only represent a rough timing analysis, more precise evaluation is performed in the following.

B. Performance Evaluation

For performance evaluation of the proposed QRD design, a wireless multi-user MISO (MU-MISO) communication system is simulated. It consists of a single base station with N_B antennas and $N_M = N_B$ decentralized non-cooperative single-antenna mobile stations. The vector of the data symbols with quadrature amplitude modulation (QAM) is given by $\mathbf{d} = [d_1, \ldots, d_{N_B}]^T$, accordingly the received symbols are defined by $\mathbf{y} = [y_1, \ldots, y_{N_M}]^T$. For the considered downlink transmission scenario, the base station sends data to the mobiles. For a fair comparison of the simulation results, a perfect knowledge of the downlink channel matrix $\mathbf{H} \in \mathbb{C}^{N_M \times N_B}$ in the base station is assumed. In this contribution, a Rayleigh multipath channel model is applied where the channel entries of $\mathbf{H}$ are normalized independently and identically distributed zero-mean complex Gaussian random variables. At the mobile stations the received signal is added to the noise vector $\mathbf{n} \in \mathbb{C}^{N_M \times 1}$ of complex Gaussian independently and identically distributed samples with variance σ_n^2. Generally, for this system the transmission equation reads as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}\,. \tag{14}$$

In order to decrease the computational effort at the receiver, pre-equalization at the transmitter is applied. Therefore, in

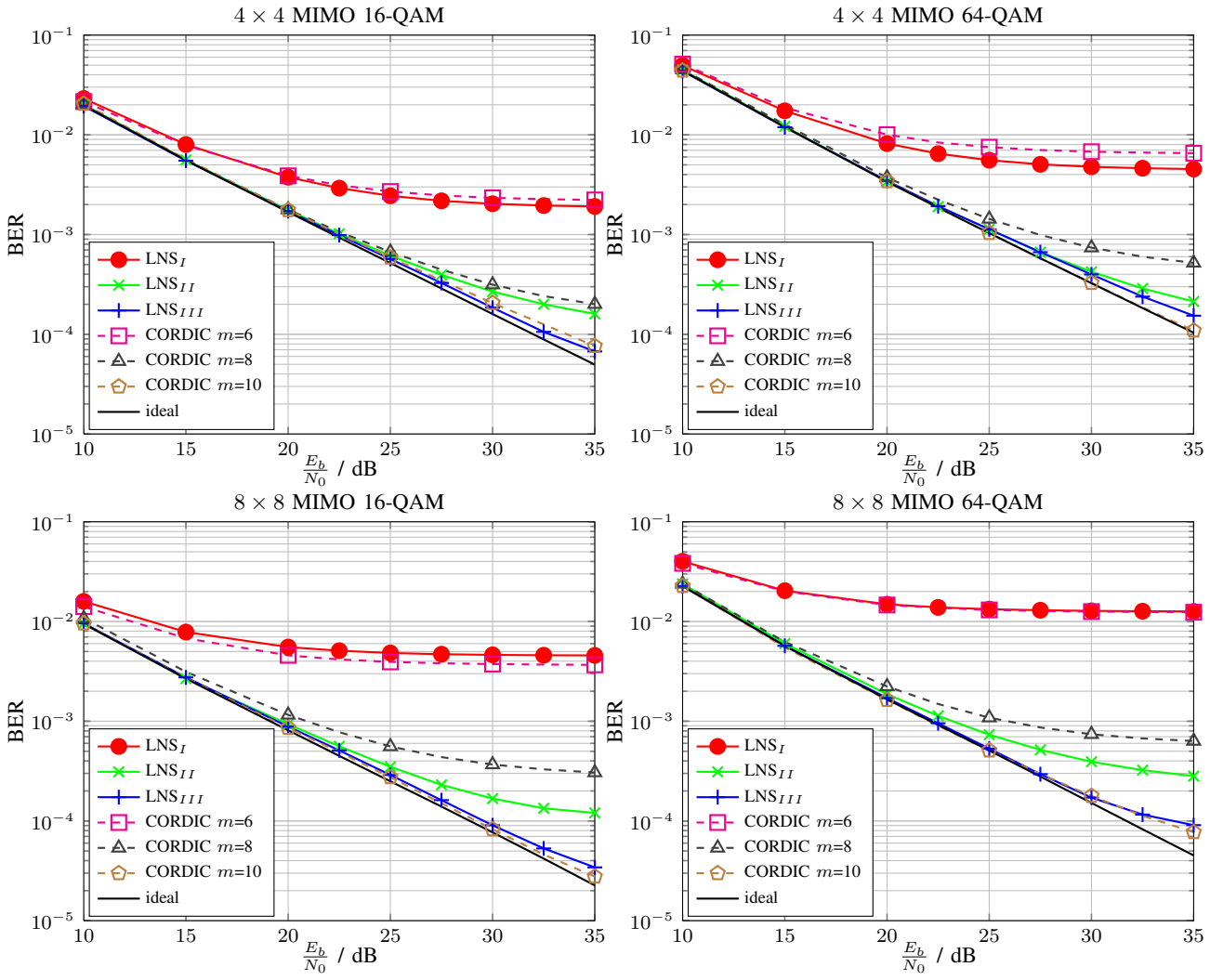

Fig. 6. Uncodes BER performance at the receivers side of the three different LNS-based QRD realizations. The CORDIC is simulated in floating-point, where m denotes the number of micro-rotations.

this contribution the Zero-Forcing THP (ZF-THP) is used, which contains the proposed LNS based QRD. The structure for the downlink THP is shown in Fig. 5. The data $\mathbf{d}$ is precoded in the base station so that the transmitted data $\mathbf{s}$ is produced. The unitary feedforward filter matrix $\mathbf{F}$ obtains spatial causality and the feedback filter matrix $(\mathbf{L} - \mathbf{I}_{N_B})$[1] eliminates the interferences. A modulo device (MOD) is used to map the values into fundamental Voronoi region which depends on the order of the chosen QAM. This will be revoked by another MOD device in the mobiles. The required filter matrices in Fig. 5 can calculated by a QRD of $\mathbf{H}^{\mathrm{H}}$, which is the conjugate transpose of the channel matrix. For a system with $N_M = N_B$ antennas this results in $\mathbf{H}^{\mathrm{H}} = \mathbf{QR}$. Then, the feedforward filter matrix is defined by $\mathbf{F} = \mathbf{QV}$, where $\mathbf{V} = \mathrm{diag}\{\mathrm{diag}^{-1}\{\mathbf{R}\}\}$ consists of the diagonal matrix elements of $\mathbf{R}$. The lower left feedback filter matrix is obtained by $\mathbf{L} = \mathbf{HQV}$. The scalar β is used to fulfill the power constraint and is defined as

$$\beta = \sqrt{\frac{N_B}{\mathrm{tr}\{\mathbf{QVV}^{\mathrm{H}}\mathbf{Q}^{\mathrm{H}}\}}}\,. \tag{15}$$

The E_b/N_0-ratio is defined as $E_b/N_0 = 1/(\log_2(M)\sigma_n^2)$ for a fair comparison of different M-QAM modulations.

The evaluation is performed for LNS$_I$, LNS$_{II}$ and LNS$_{III}$ with 16-QAM and 64-QAM modulation schemes as well as 4×4 and 8×8 multi-antenna scenarios. The resulting uncoded Bit Error Rates (BER) over E_b/N_0 are compared to ideal and CORDIC-based QRD designs. As it turns out, the LNS$_I$, LNS$_{II}$ and LNS$_{III}$ reach better accuracy nearly in all investigated scenarios than the floating-point CORDIC designs with six, eight and ten iteration steps, respectively. A detailed overview is given in Fig. 6.

C. IC Implementation

In order to achieve a detailed evaluation, the proposed LNS-based QRD designs LNS$_I$ and LNS$_{III}$ are implemented in

[1] $\mathbf{I}_{N_B}$ denotes the $N_B \times N_B$ identity matrix.

TABLE II
IC POST-ROUTE SYNTHESIS RESULTS OF LNS_I AND LNS_{III} FOR A 4×4 MIMO SYSTEM COMPARED TO ACTUAL REFERENCES.

Reference	Configuration	Algorithm	Technology [nm]	Frequency [MHz]	Area [kGE]	Power [mW]	Throughput [MQRDs/s]
This work	**LNS_I**	**Givens rotation**	**130**	**145**	**17.93**	**5.68**	**0.51**
This work	**LNS_{III}**	**Givens rotation**	**130**	**129**	**22.38**	**5.73**	**0.45**
[13]	CORDIC	Givens rotation	180	272	-	105	0.71
[14]	CORDIC	Givens rotation	180	100	111	319	25
[15]	CORDIC	modified Gram-Schmidt	180	166	48.7	-	2.08
[16]	CORDIC	Givens rotation	130	270	36	-	6.76

CMOS for a 4×4 MIMO system. Logical and physical synthesis is performed, considering timing back-annotation as well as parasitic effects. As target technology the UMC-Faraday 130 nm process is chosen. In order to enable a fair complexity comparison, kilo gate equivalents (kGE) are regarded. LNS_I and LNS_{III} reach maximum frequencies of 145 MHz and 129 MHz and kGEs of 17.93 and 22.38, respectively. Evaluating the power consumption, our approach achieves $5.68\,\mu$W and $5.73\,\mu$W. A detailed overview also taking actual references into account is given in Tab. II. An important aspect that must be discussed is the throughput performance of our work. Thus, actual references as e.g. given in [14] obtain significantly better QRD processing results. This is explainable by the implementation style that we chose in this first approach in order to generally highlight advantages of LNS-based QRD processing. Thus, our proposal could be used in mobile communication applications with relaxed timing conditions. In order to speed-up the QRD of out approach, hardware implementation techniques, such as pipelining or replication [11] must be installed, which will evidently increase the throughput. So this has to be a major concern of future work.

VI. CONCLUSION

In this paper a QR-decomposition module based on logarithmic number system calculation is introduced. Its main advantage refers to the simplification of complex calculations to additions, subtractions or shifts. For number format conversion, low complexity transformation units are installed that use nonuniform and multiplier-less piecewise function approximation. Simple equation transformation of the Givens rotation is performed in order to keep all calculations in a convenient range. For simulation, three different converter approximations have been considered. The comparison with default CORDIC-based designs proved this approach to be more efficient regarding the total number of adders. The IC implementation achieved good results in terms of complexity and power consumption. For future work, the throughput must be raised as only a comparatively low performance is obtained. However, this is solvable e.g. by the application of well-known hardware implementation techniques.

REFERENCES

[1] D. Wübben and K.-D. Kammeyer, "Low Complexity Successive Interference Cancellation for Per-Antenna-Coded MIMO-OFDM Schemes by Applying Parallel-SQRD," in *IEEE 63rd Vehicular Technology Conference, 2006. VTC 2006-Spring*, vol. 5, May 2006, pp. 2183 –2187.

[2] R. F. H. Fischer, *Precoding and Signal Shaping for Digital Transmission*. John Wiley & Sons, Inc., 2005.

[3] J. Detrey and F. de Dinechin, "A VHDL library of LNS operators," in *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, vol. 2, Nov. 2003, pp. 2227 – 2231 Vol.2.

[4] D. De Caro, N. Petra, and A. Strollo, "Direct Digital Frequency Synthesizer Using Nonuniform Piecewise-Linear Approximation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 10, pp. 2409–2419, Oct. 2011.

[5] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Johns Hopkins Univ. Press, 1996.

[6] J. Volder, "The CORDIC computing technique," in *Papers presented at the the March 3-5, 1959, western joint computer conference*, ser. IRE-AIEE-ACM '59 (Western). New York, NY, USA: ACM, 1959, pp. 257–261.

[7] J. N. Mitchell, "Computer Multiplication and Division Using Binary Logarithms," *IRE Transactions on Electronic Computers*, vol. EC-11, no. 4, pp. 512–517, Aug. 1962.

[8] J.-M. Muller, *Elementary functions : Algorithms and Implementation*, 2nd ed. Birkhaeuser Boston, 2006.

[9] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*. Oxford University Press, 2010.

[10] M. Lyons, M. Hempstead, G.-Y. Wei, and D. Brooks, "The accelerator store framework for high-performance, low-power accelerator-based systems," *Computer Architecture Letters*, vol. 9, no. 2, pp. 53 –56, Feb. 2010.

[11] H. Kaeslin, *Digital Integrated Circuit Design: From VLSI Architectures to CMOS Fabrication*. Cambridge University Press, 2008.

[12] M. Keating, *Low power methodology manual: for system-on-chip design*, corr. 2. print. ed., ser. Series on integrated circuits and systems. New York, NY: Springer, 2008.

[13] C. Studer, P. Blosch, P. Friedli, and A. Burg, "Matrix Decomposition Architecture for MIMO Systems: Design and Implementation Trade-offs," in *Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers, 2007. ACSSC 2007.*, Nov. 2007, pp. 1986 –1990.

[14] Z.-Y. Huang and P.-Y. Tsai, "Efficient Implementation of QR Decomposition for Gigabit MIMO-OFDM Systems," *Regular Papers, IEEE Transactions on Circuits and Systems I*, vol. 58, no. 10, pp. 2531 –2542, Oct. 2011.

[15] P. Luethi, C. Studer, S. Duetsch, E. Zgraggen, H. Kaeslin, N. Felber, and W. Fichtner, "Gram-Schmidt-based QR decomposition for MIMO detection: VLSI implementation and comparison," in *IEEE Asia Pacific Conference on Circuits and Systems, 2008. APCCAS 2008.*, Dec. 2008, pp. 830 –833.

[16] D. Pathel, M. Shabany, and P. Genn Gulak, "A low-complexity high-speed QR decomposition implementation for MIMO receivers," in *IEEE International Symposium on Circuits and Systems, 2009*, 2009.